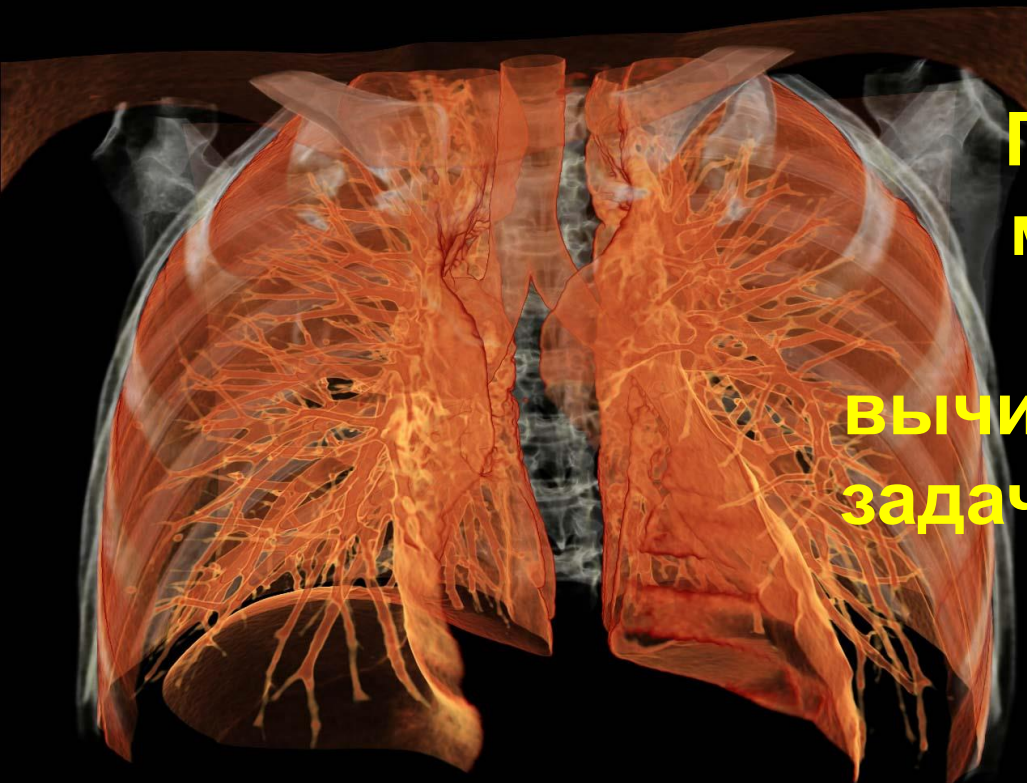




Шестая международная молодежная научно-практическая школа  
**Высокопроизводительные вычисления на Grid системах**



**Применение лучевых методов как подход к параллельным вычислениям на примере задач научно-инженерной визуализации**

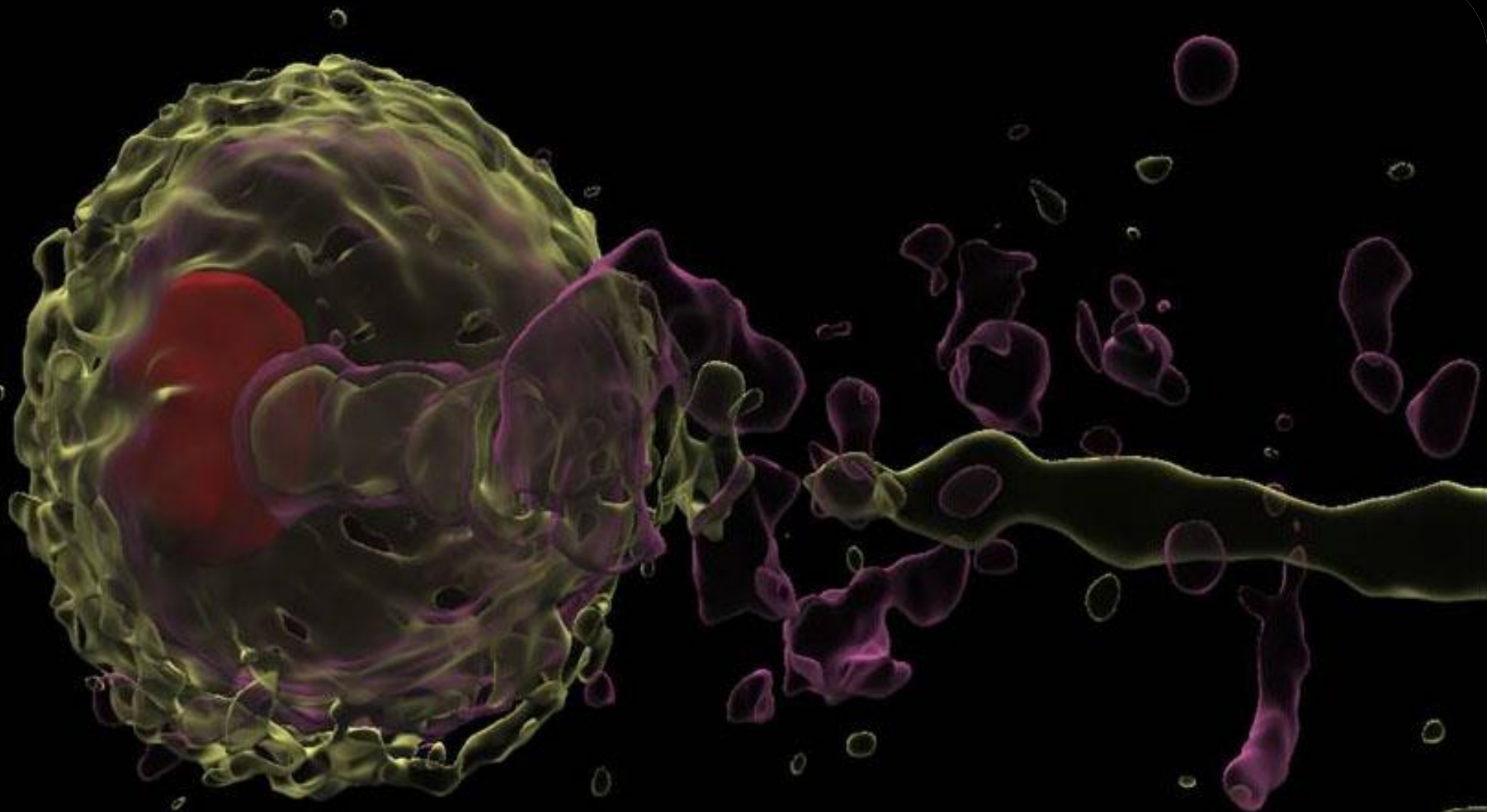
Профессор кафедры математического обеспечения ЭВМ ННГУ им. Н.И. Лобачевского, руководитель лаборатории компьютерной графики и мультимедиа  
**Турлапов Вадим Евгеньевич**, [vadim.turlapov@gmail.com](mailto:vadim.turlapov@gmail.com)



**Примеры решенных задач  
научной 3D визуализации**

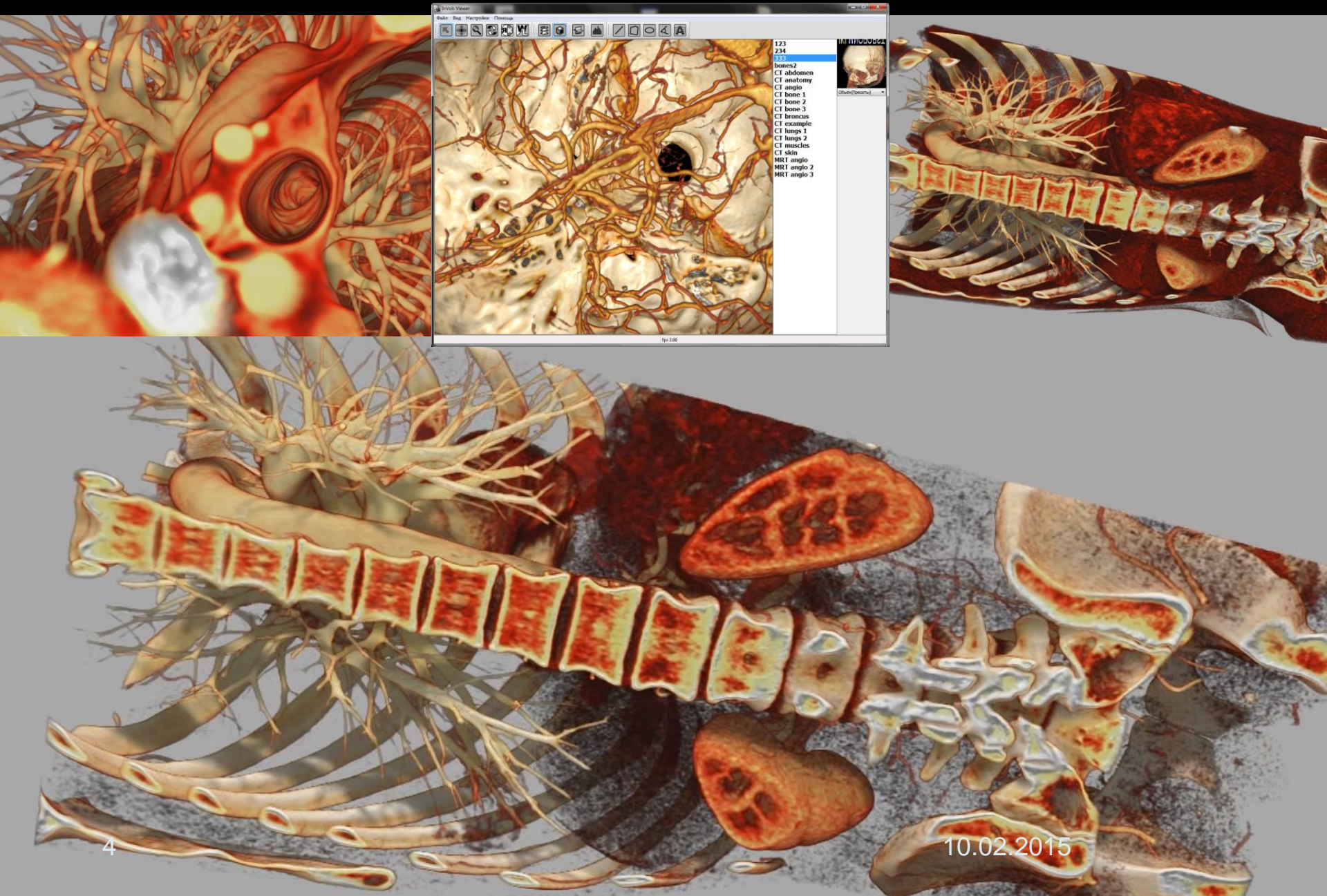
# Научная 3D визуализация.

## Фемтосекундный лазерный импульс в плазме



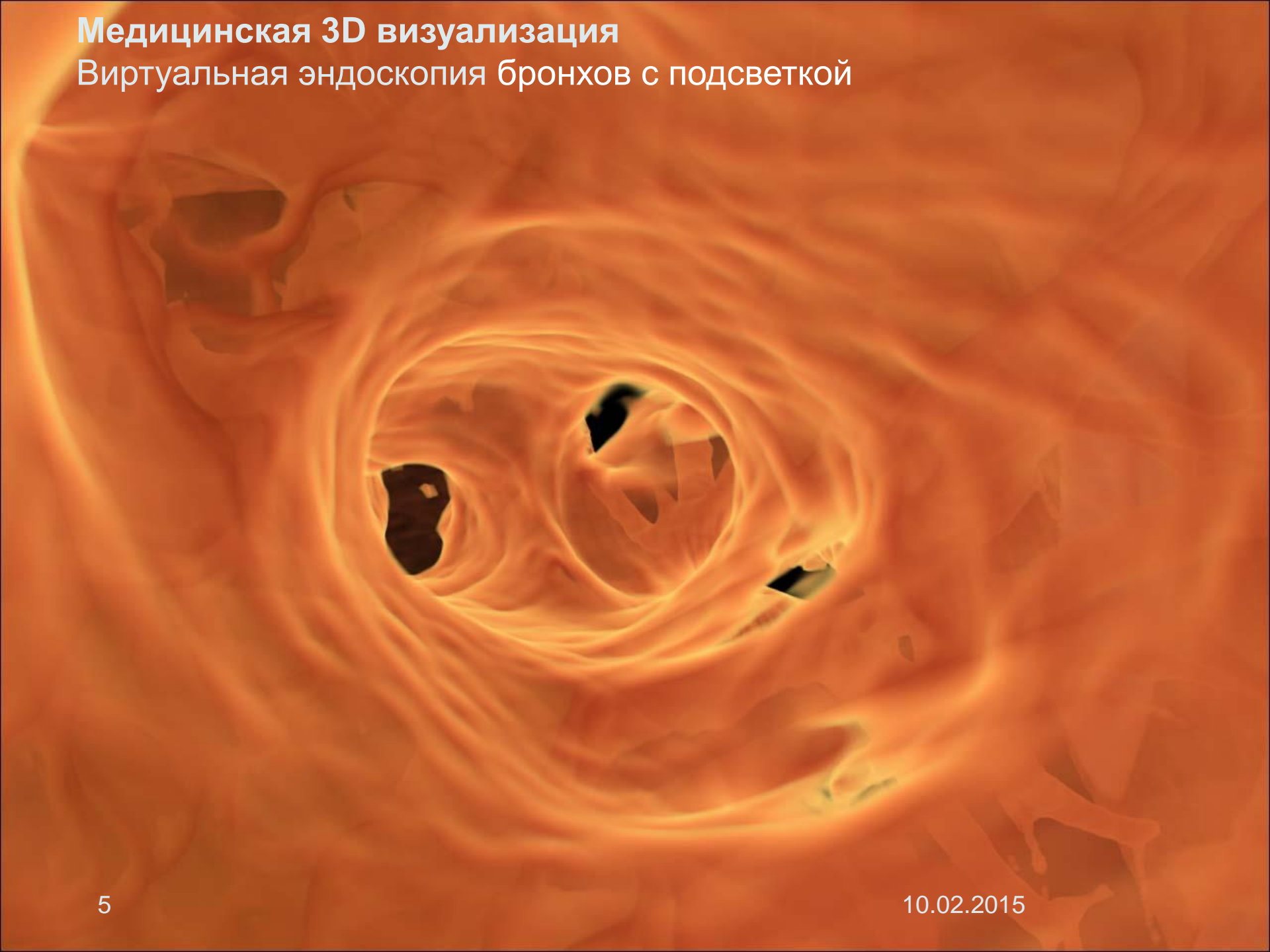


# Медицинская 3D визуализация (с тенями, в реальном времени)



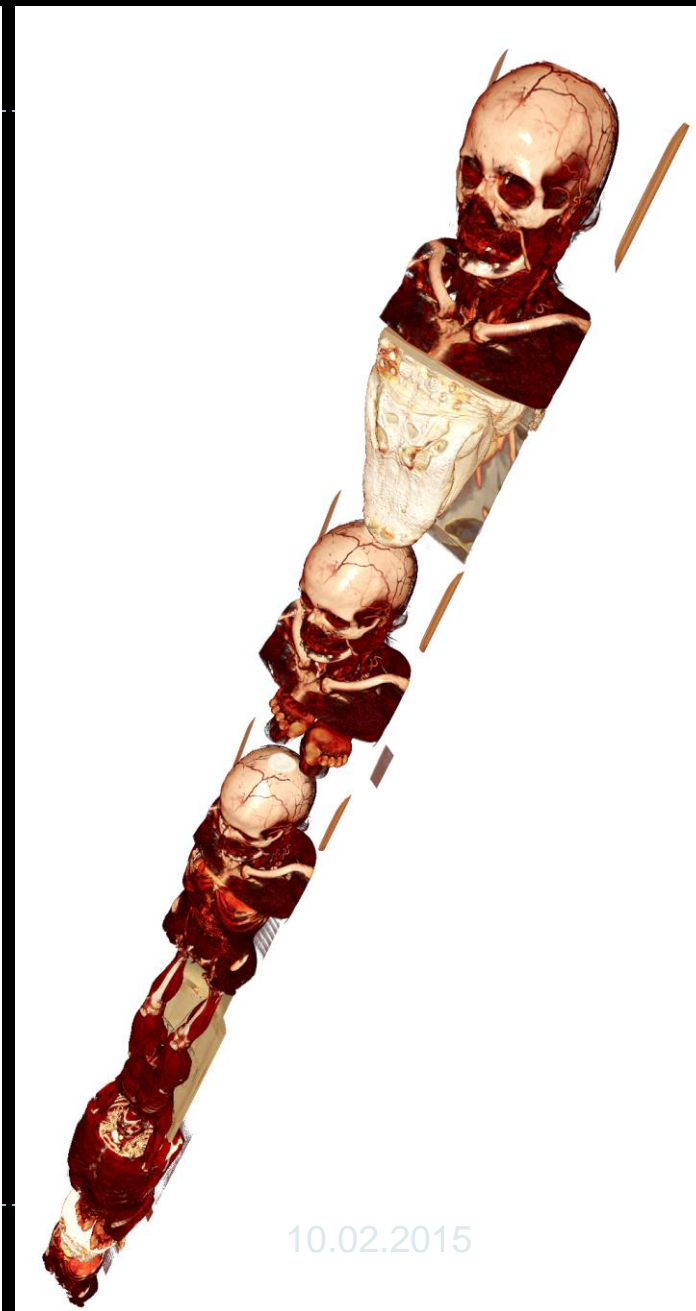
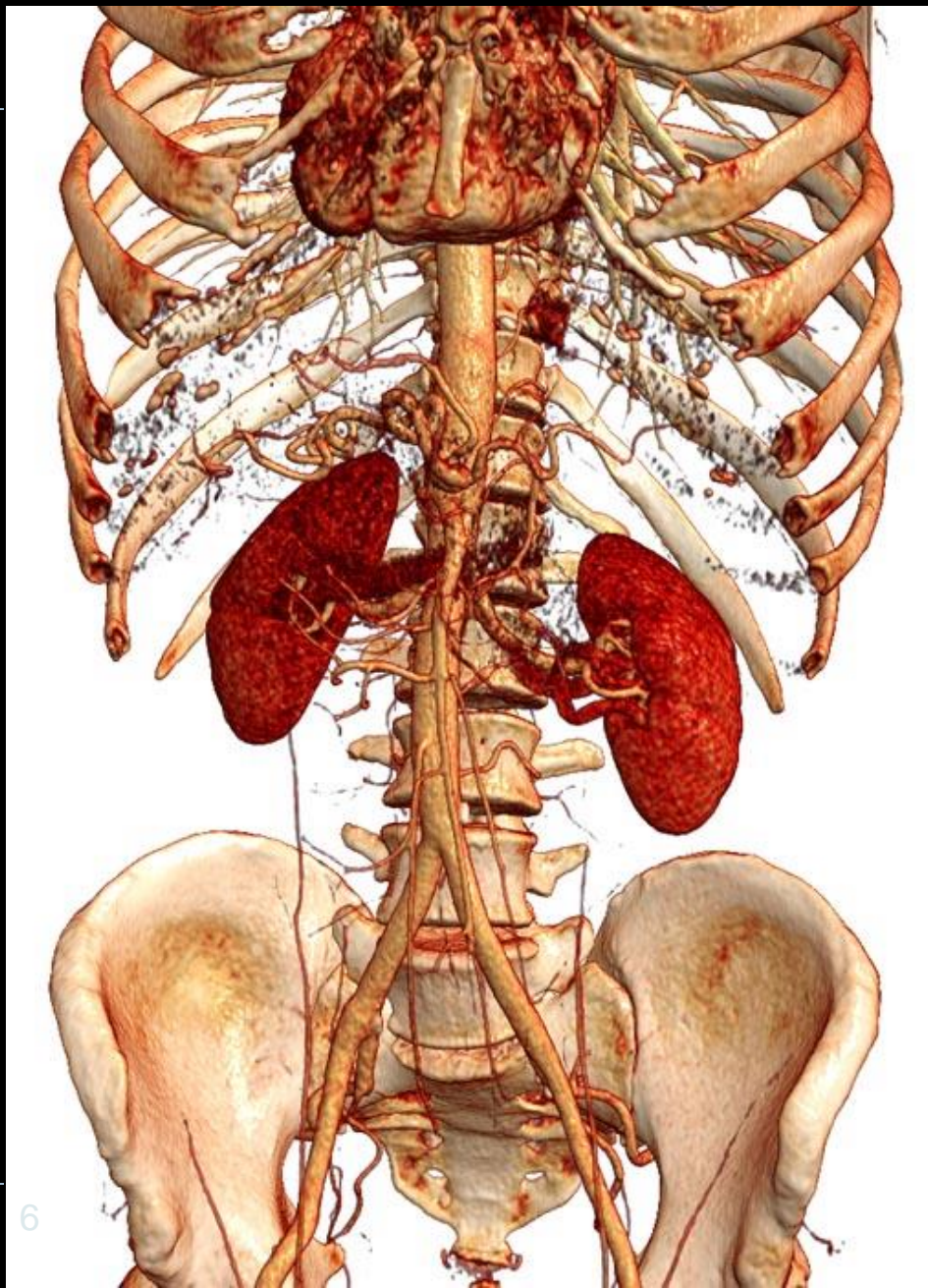
# Медицинская 3D визуализация

## Виртуальная эндоскопия бронхов с подсветкой





# Медицинская 3D визуализация. Рекордная производительность и качество



10.02.2015

# Хранилище томограмм на кластере ННГУ

## Облачный Интернет-сервис 2D-3D визуализации томограмм



---

# За что я люблю массивно-параллельные (massive parallel) архитектуры



# Два закона (Амдала и Густафсона) и одно общее следствие

**Закон Амдала (Amdahl's law, 1967):**

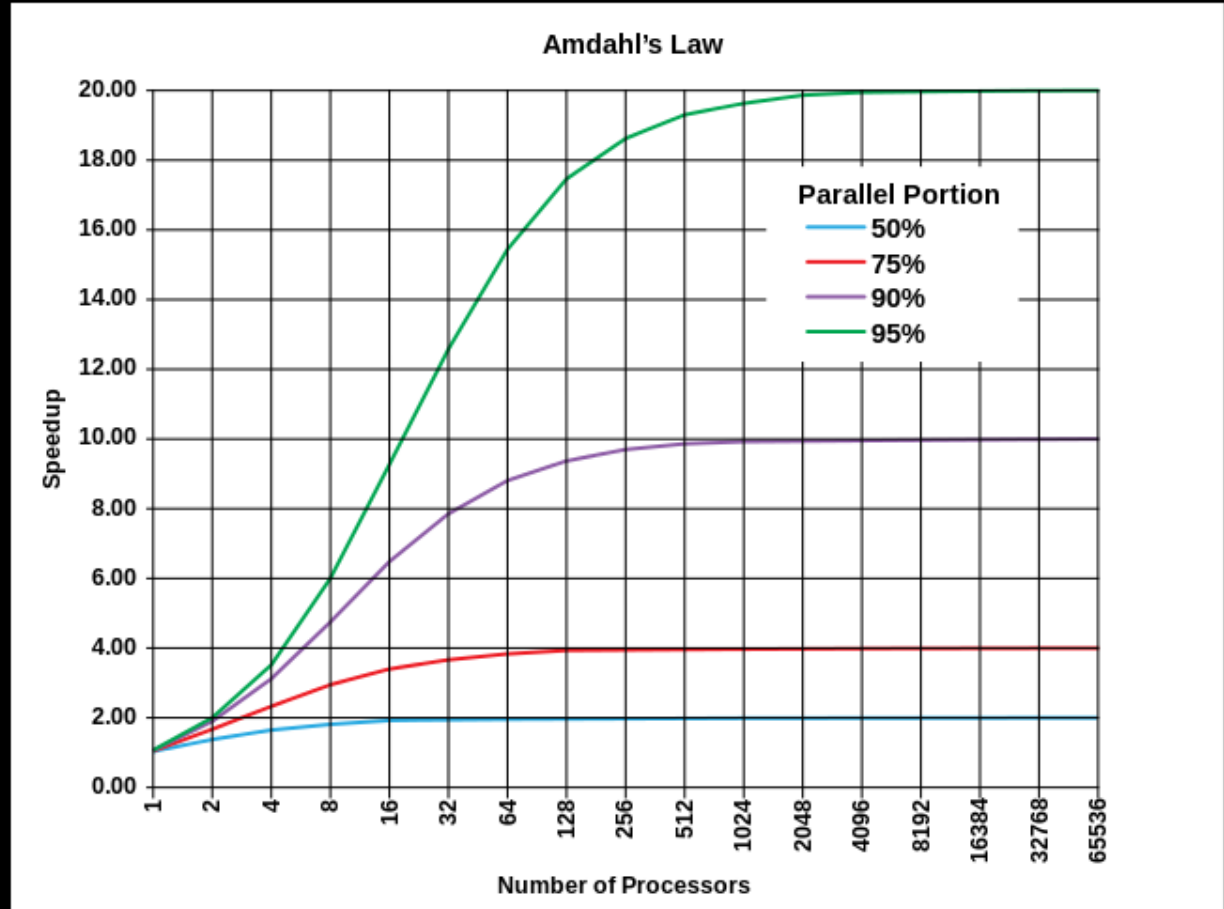
$$S(n) = \frac{1}{f + (1-f)/n}$$

где  $n$  – число процессоров,  $S$  – ускорение,  $f$  – часть кода, не поддающаяся распараллеливанию. → часто не стоит наращивать  $n$ , но стоит снизить  $f$

**Закон Густафсона-Барсиса (Gustafson – Barsis's law, 1988:** пользователи стремятся не сократить время работы текущей версии задачи, а перейти к новой версии, обеспечивающей новый уровень параллельности решения):

$$S(n) = n - f \cdot (n - 1)$$

→ всегда стоит наращивать  $n$  и снижать  $f$



"AmdahlsLaw" by Daniels220 at English Wikipedia - Own work based on: File:AmdahlsLaw.png. Licensed under CC BY-SA 3.0 via Wikimedia Commons - <http://commons.wikimedia.org/wiki/File:AmdahlsLaw.svg#mediaviewer/File:AmdahlsLaw.svg>

# Два закона (Амдала и Густафсона) и одно общее следствие

## Закон Густафсона-Барсиса

(*Gustafson – Barsis's law*, 1988: перейти к новой версии, обеспечивающей новый уровень оптимизации и параллельности решения):

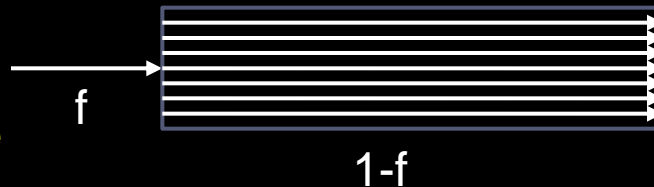
$$S(n) = n - f \cdot (n - 1)$$

n	95%	90%	75%	50%
1	1	1	1	1
2	1.95	1.9	1.75	1.5
4	3.85	3.7	3.25	2.5
8	7.65	7.3	6.25	4.5
16	15.25	14.5	12.25	8.5
32	30.45	28.9	24.25	16.5

**Почему так? Что произошло?**

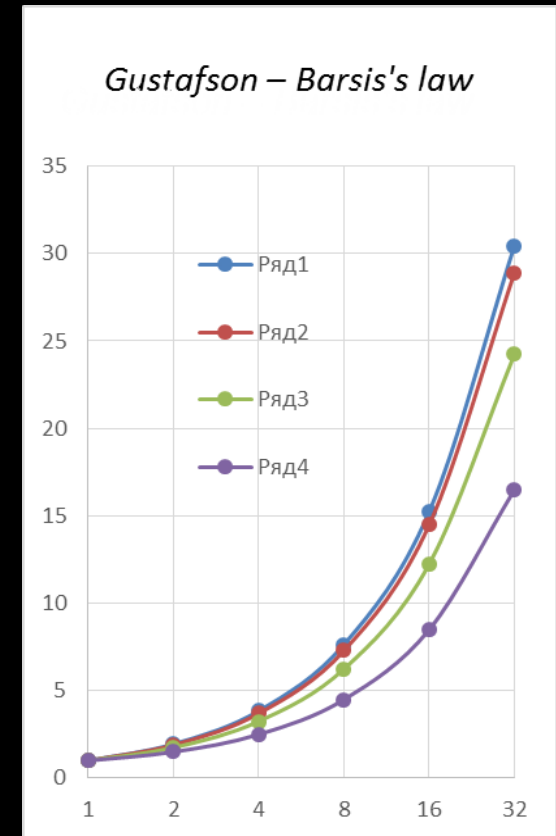
**Ответ:**

**Последовательная часть  $f$  выполняется в каждом потоке**



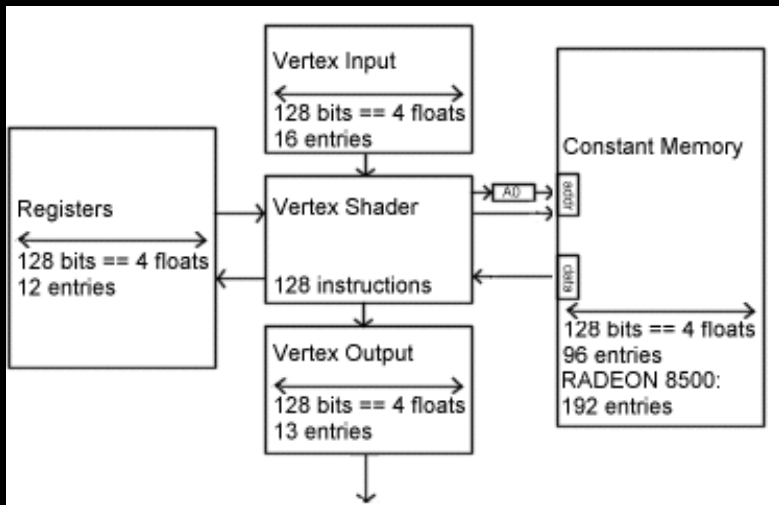
**Следствие:**

**Всегда наряду с оптимизацией следует сводить алгоритм к массивно-параллельному (т.е. с нулевым  $f$  в смысле закона Амдала)**

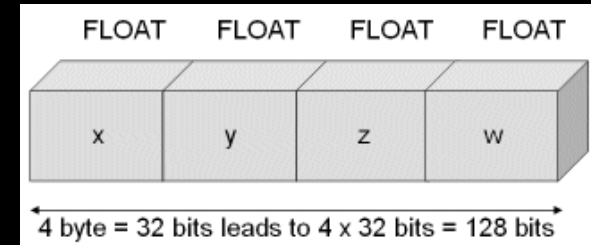


# Архитектура GPU и параллельное программирование

## Vertex Shader 128bit Architecture



## Специализация регистров шейдера



Registers:	Number of Registers	Properties
<b>Input (v0 - v15)</b>	<b>16</b>	<b>RO</b>
<b>Output (o*)</b>	<b>GeForce 3/4TI: 9; RADEON 8500: 11</b>	<b>WO</b>
<b>Constants (c0 - c95)</b>	<b>vs.1.1 Specification: 96; RADEON 8500: 192</b>	<b>RO</b>
<b>Temporary (r0 - r11)</b>	<b>12</b>	<b>RW</b>
<b>Address (a0.x)</b>	<b>1 (vs.1.1 and higher)</b>	<b>WO (W: only with mov)</b>

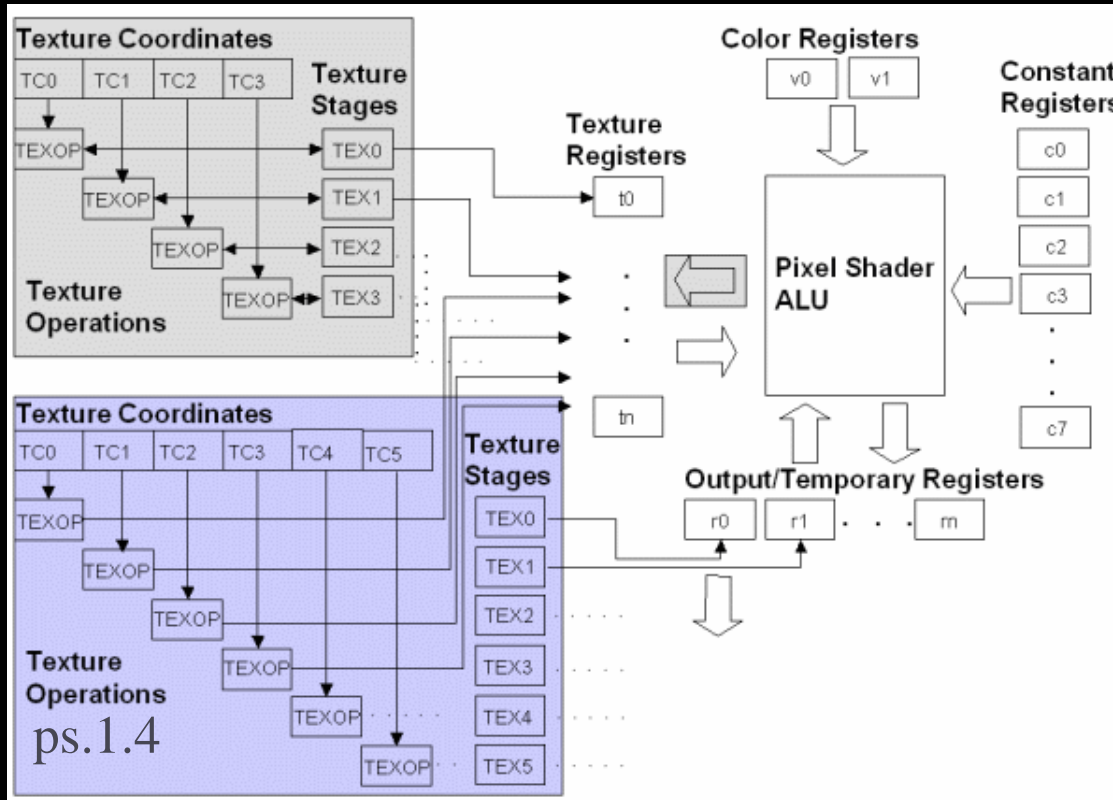
### Выходные регистры вершинного шейдера:

Name	Value	Description
oDn	2 quad-floats	Цвет для пиксельного шейдера diffuse (oD0) и specular (oD1).
oPos	1 quad-float	Output position в однородном усеченном пространстве
oTn	до 8 quad-floats	Выходные координаты текстуры
oPts.x	1 scalar float	Размер точки (скаляр)
oFog.x	1 scalar float	Коэффициент затуманивания



# Архитектура GPU и параллельное программирование

## Pixel Shader 128bit Architecture



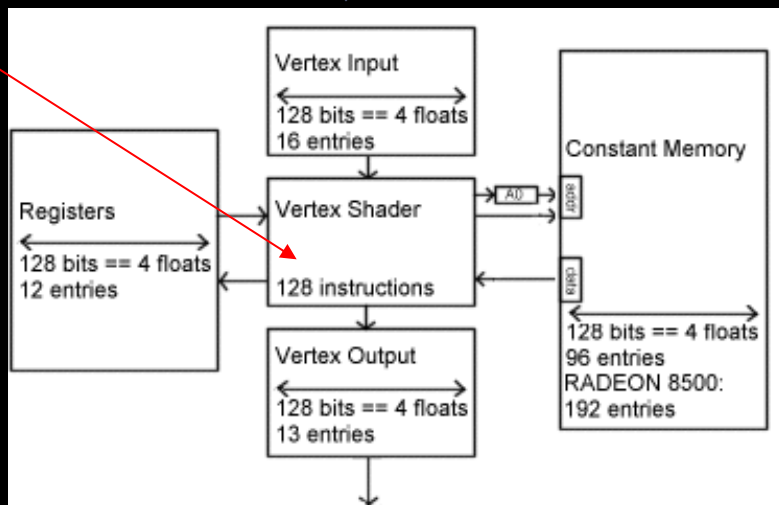
Type	Name	ps.1.1	Read/Write
Constant Registers	$c_n$	8	RO
Texture Registers	$t_n$	4	RW → RO
Temporary Registers	$r_n$	2 → n	RW
Color Registers	$v_n$	2	RO

# Архитектура GPU и параллельное программирование

Данные уникальные для каждой вершины **attribute**-переменные (RO)

Окно исполнения:  
короткие адреса,  
несколько команд в 1 регистре

(RW)



Константы и **uniform**-переменные общие для всех потоков (RO)

Аппаратная линейная интерполяция

**varying**-переменные

Грубая оценка веса одного потока –  
**32 регистра** →  
**Много потоков** →  
**Эффективный кэш**

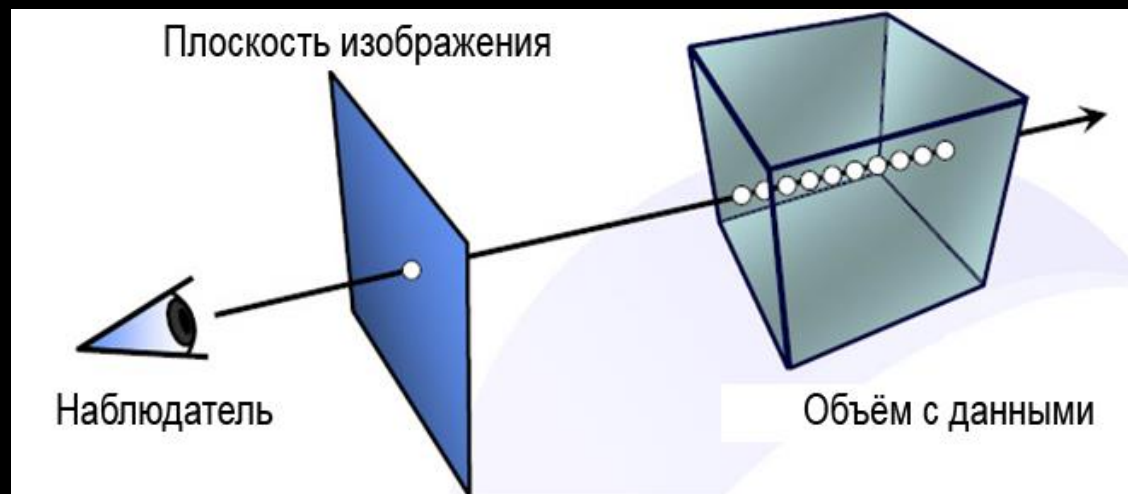
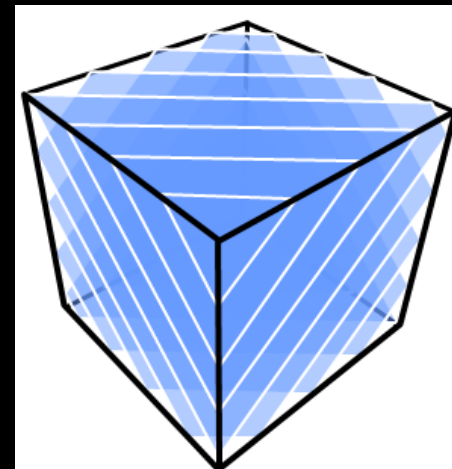
---

# Лучевые методы в задаче научной визуализации



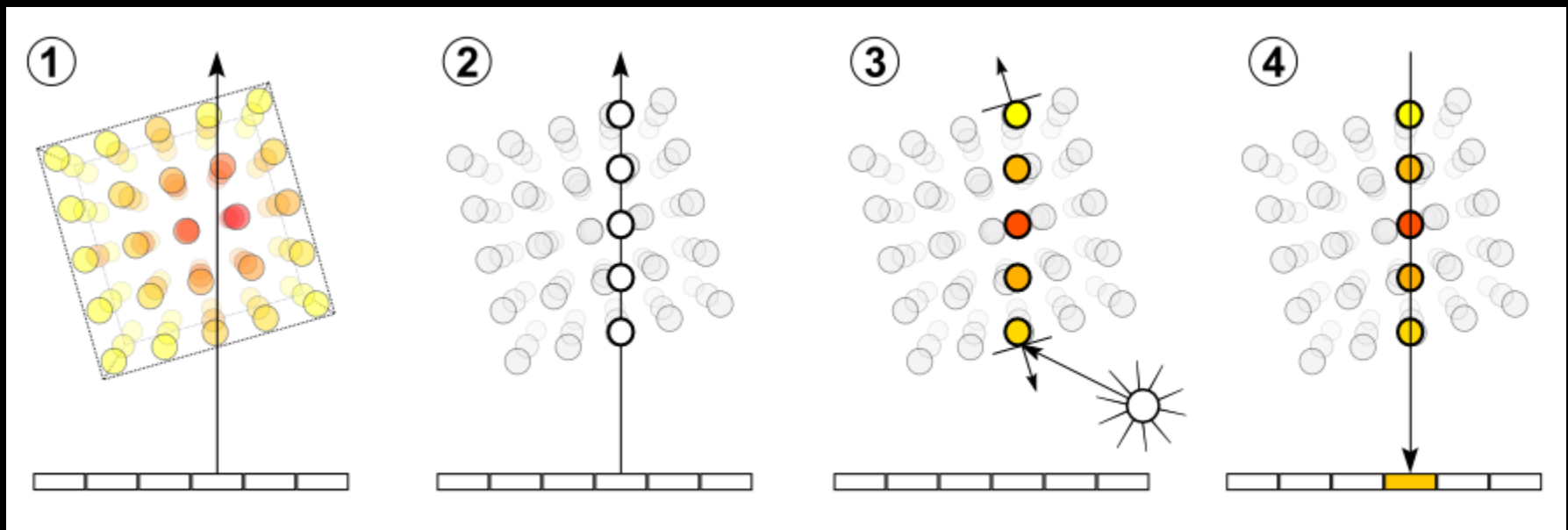
# Методы объёмного рендеринга

- ▶ **Метод слоёв**
  - ▶ Экономия памяти. Производительность
- ▶ **Метод обратной трассировки лучей (Ray Casting)**
  - ▶ качество
  - ▶ параллельность



# Трассировка луча через регулярную сетку

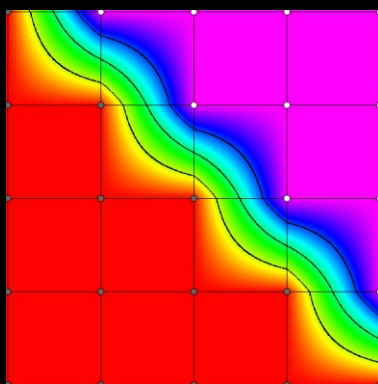
1. Генерация луча для каждого пикселя окна экрана
2. Конечное число шагов вдоль луча
3. Интегрирование цвета и непрозрачности
4. Моделирование освещенности



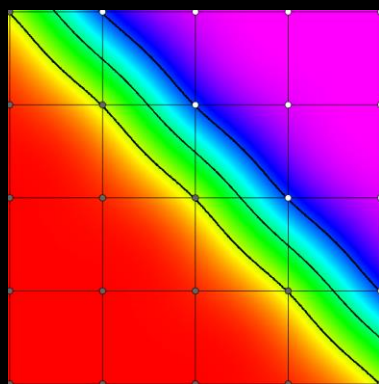
# Процесс визуализации объёмных данных всегда включает в себя этапы *интерполяции* и *классификации*

## 1. Интерполяция

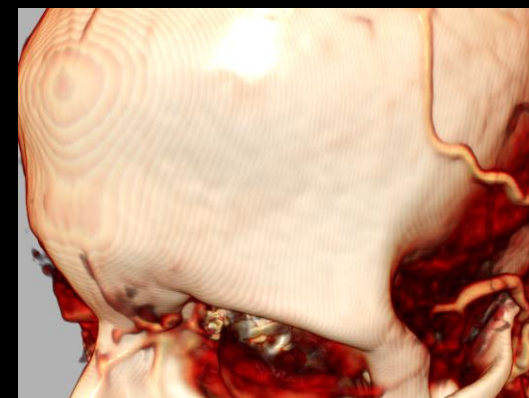
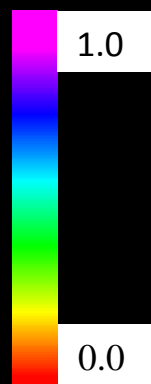
Ступенчатые дефекты визуализации, возникающие на данном этапе, можно значительно уменьшить с помощью трикубической интерполяции



Билинейная



Бикубическая



Трилинейная

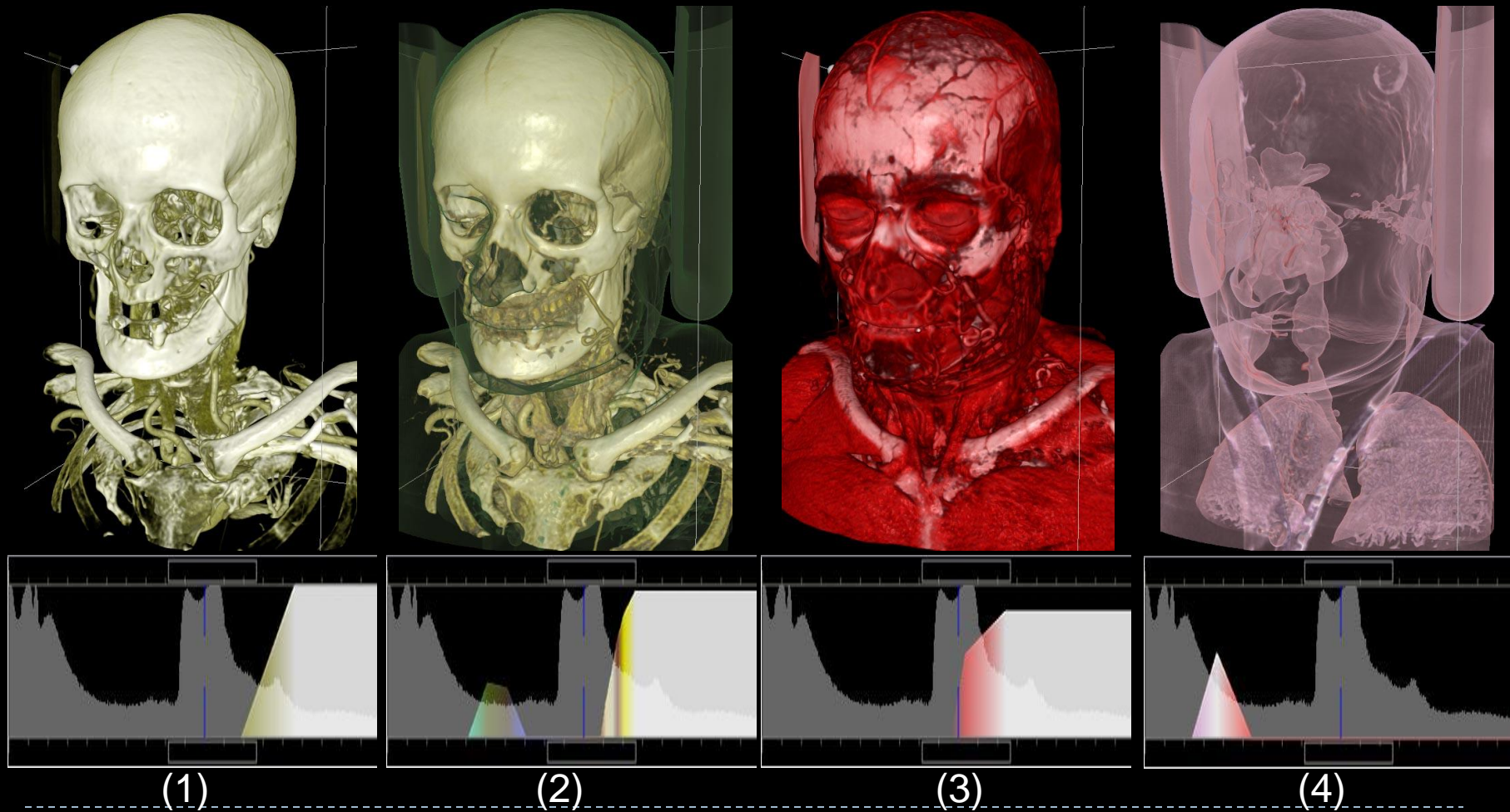


Трикубическая



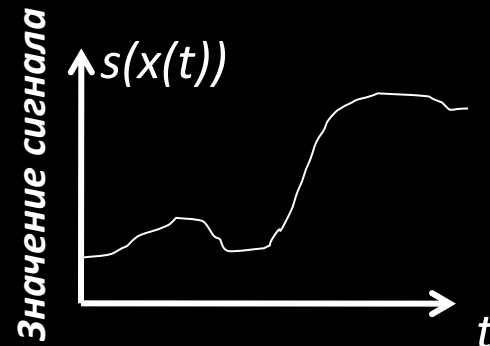
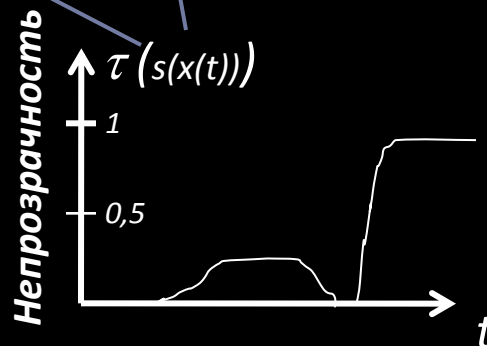
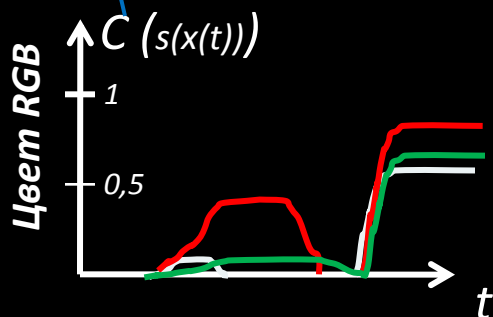
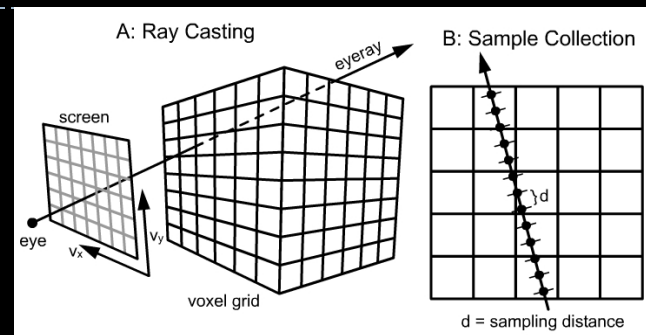
Процесс визуализации объёмных данных  
всегда включает в себя этапы *интерполяции* и  
*классификации*

2. Классификация преобразует значения (s) в цвет через *Transfer Function*



# Математическая модель трассировки луча: интеграл объемного рендеринга

$$I = \int_0^{t_E} C(s(\mathbf{x}(t))) \times \tau(s(\mathbf{x}(t))) \times \exp\left(-\int_0^t \tau(s(\mathbf{x}(t'))) dt'\right) dt$$



Пример зависимостей различных величин в подынтегральной функции

$C(s(\mathbf{x}))$  – постклассифицированный цвет для значения поля  $s$  в точке  $\mathbf{x}$

$\tau(s(\mathbf{x}))$  – непрозрачность для значения поля  $s$  в точке  $\mathbf{x}$

Интеграл описывает процесс накопления (интегрирования) цвета вдоль луча  $\mathbf{x}(t)$  с учетом его затухания в зависимости от расстояния  $t$  до точки наблюдения

# Процедура численного интегрирования в модели с постклассификацией

---

Пусть отрезок интегрирования разбивается на  $n$  элементарных отрезков равной длины  $d = D / n$

$$\begin{aligned} \exp\left(-\int_0^t \tau(s(\mathbf{x}(t'))))dt'\right) &\approx \exp\left(-\sum_{i=0}^{t/d} \tau(s(\mathbf{x}(id)))d\right) \\ &= \prod_{i=0}^{t/d} \exp(-\tau(s(\mathbf{x}(id)))d) \approx \prod_{i=0}^{t/d} (1 - \alpha_i) \end{aligned} \quad (1)$$

$$\alpha_i = 1 - \exp(-\tau(s(\mathbf{x}(id)))d) \approx \tau(s(\mathbf{x}(id)))d \quad (2)$$

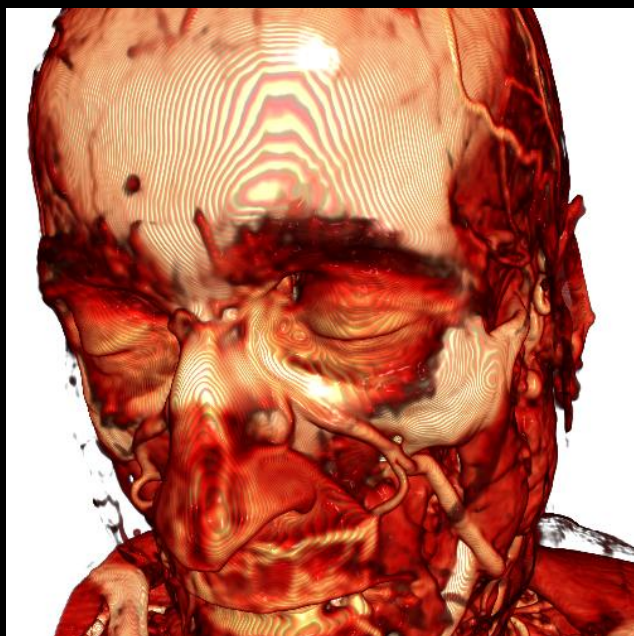
$$I \approx \sum_{i=0}^n C_i \cdot \alpha_i \prod_{j=0}^{i-1} (1 - \alpha_j) \quad (3)$$

$C_i$   $\alpha_i$  – цвет и непрозрачность  $i$ -ого сегмента луча;  $n$  – число шагов луча

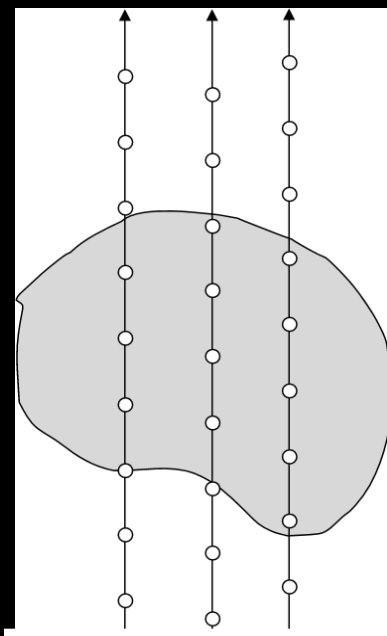
# Случайный сдвиг (*jittering*) стартовых позиций лучей для устранения регулярности дефектов постклассификации

---

Цвет каждого пикселя изображения рассматривается, как случайная величина.



Старт лучей общей плоскости



Случайный сдвиг луча (*jittering*)

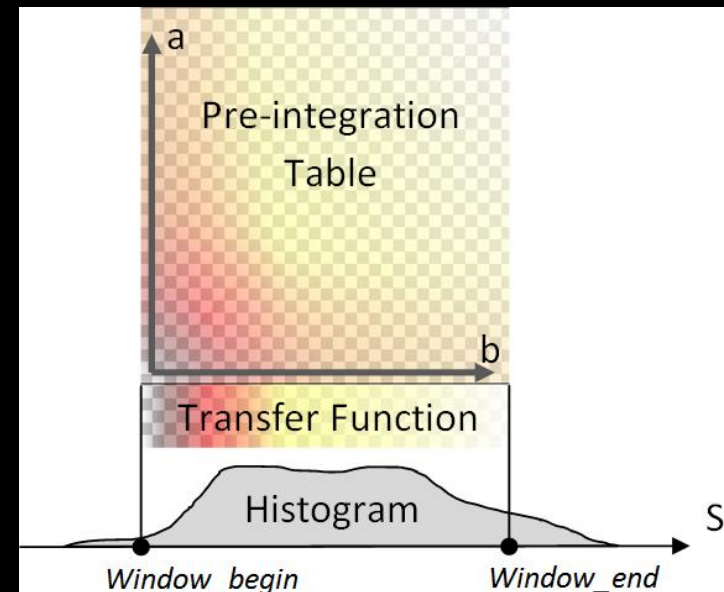
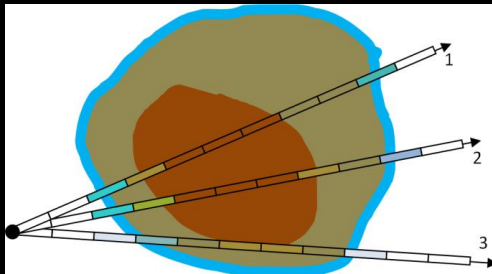
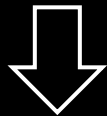
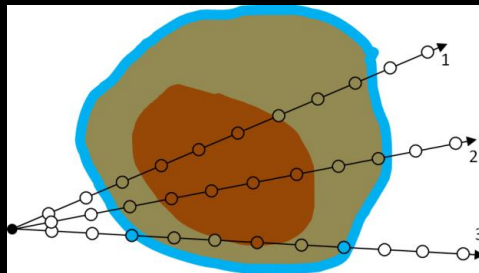


Старт со случайным сдвигом



# Интегрирование с использованием таблиц. Предынтегрированный объёмный рендеринг (PDVR)

1. Любой отрезок по  $s$  характеризуется двумя значениями данных – выборки на его концах.
2. Цвета отрезков хранят в предынтегрированной таблице, вычисленной заранее на CPU (декомпозиция интегрирования - взятие интеграла по частям)
3. При классификации PDVR использует цвета отрезков по  $s$ , а не точек (предынтегрированная классификация)



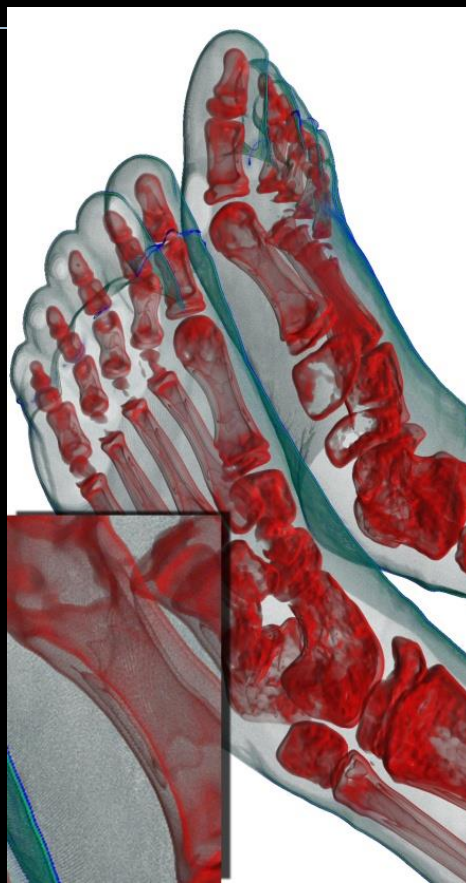
Transfer Function и соответствующая ей предынтегрированная таблица



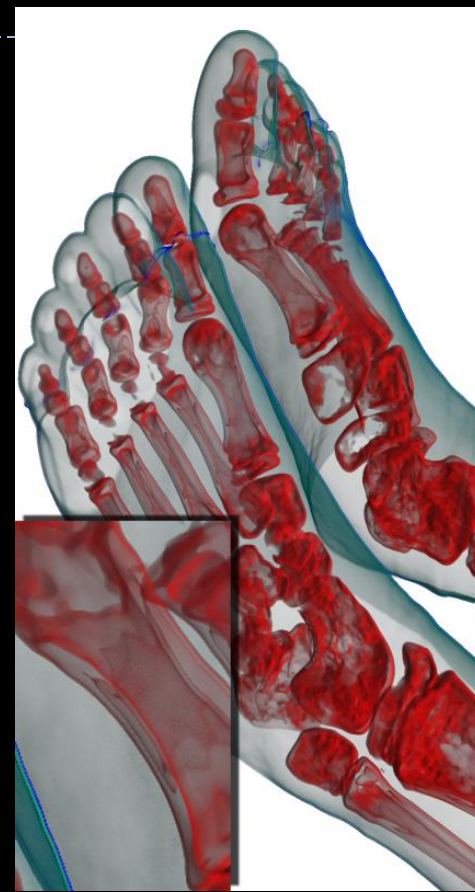
# Улучшение качества при использовании PDVR



один кадр обычного  
рендеринга (UDVR)  
 $PSNR=12dB$



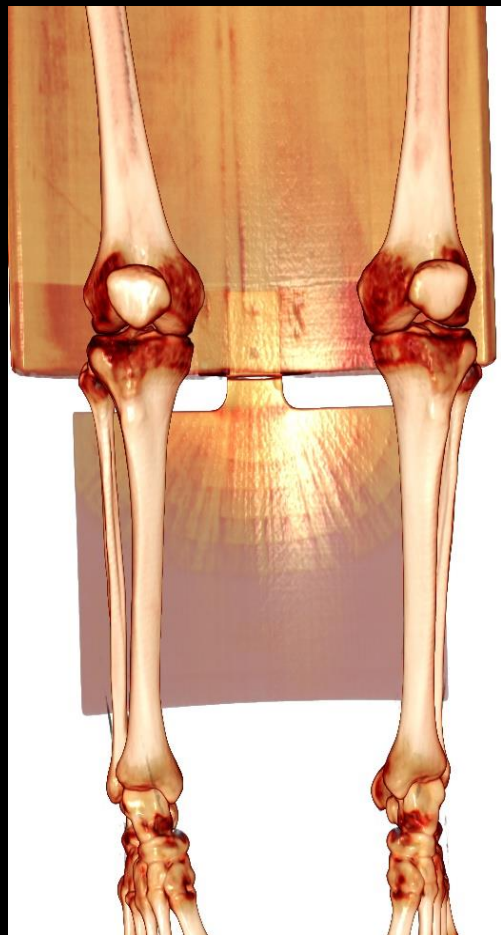
усреднение 60 кадров  
обычного рендеринга  
 $PSNR=27dB$



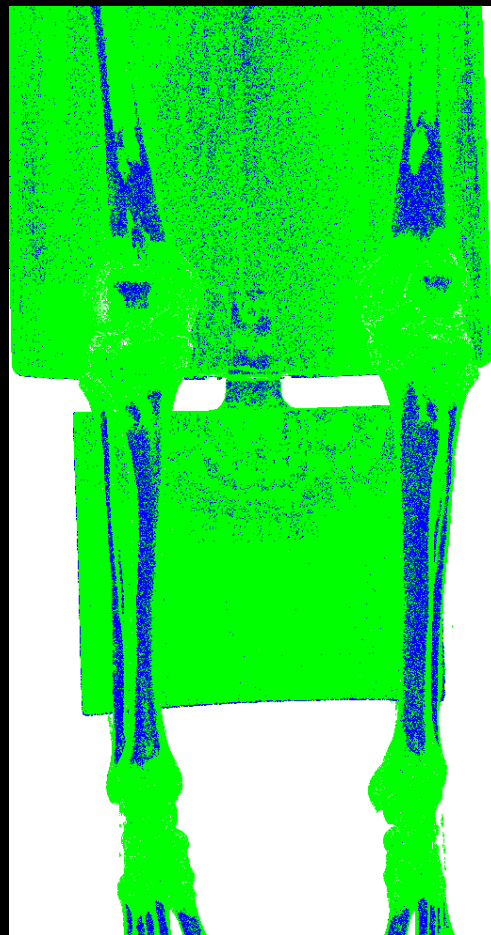
предынтегрированный  
рендеринг (PDVR)  
 $PSNR=36dB$

Частота выборки: 2

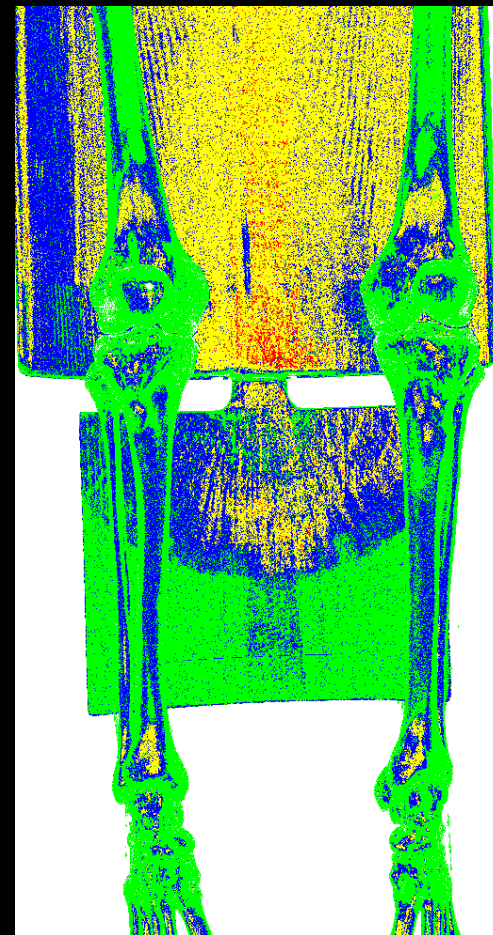
# Недостаток предынтегрированного рендеринга: дефекты изображения при использовании локального освещения



Результат рендеринга;

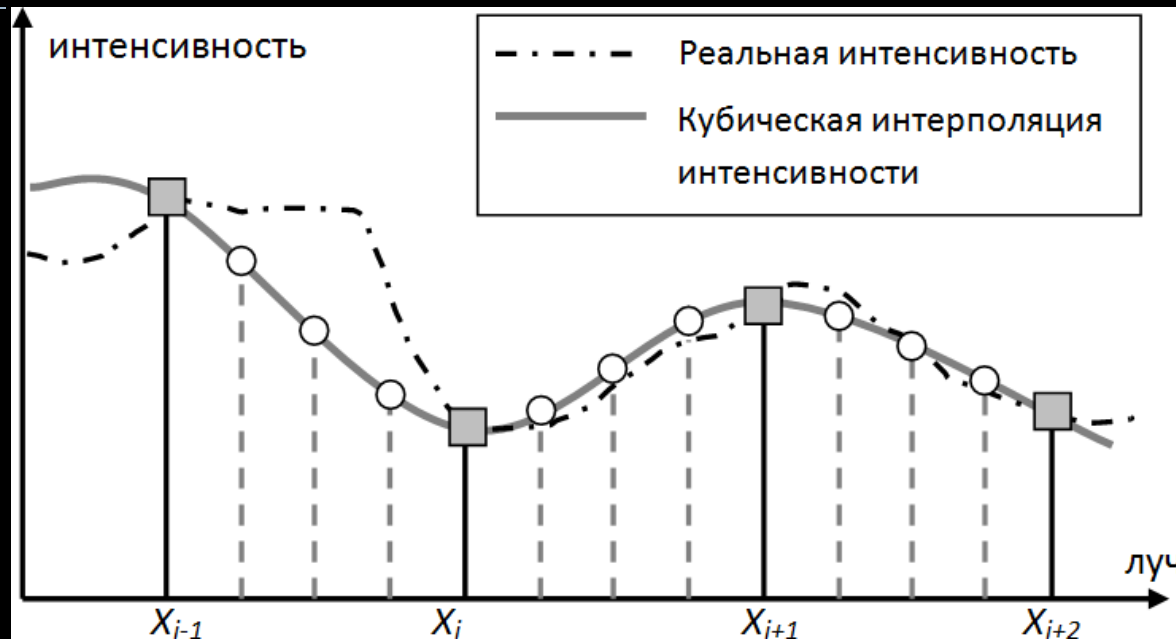


Без освещения  
PSNR = 36.4dB



С освещением  
PSNR = 22.1dB

# Метод виртуальных выборок с предынтегрированием



Известный метод виртуальных выборок дополнен заменой постклассификации на предынтегрированную классификацию



# Количественная оценка качества визуализации

- 1) Исследованы оценки качества (дефектов) изображения, принятые в теории сигналов, 2D и 3D сжатии, в оценке качества видео (PSNR, SNR, SSIM)
- 2) Для задачи количественной оценки дефектов постклассификации в объёмной визуализации предлагается использовать логарифмическую оценку, подобную PSNR

Дисперсия цвета пикселя:

$$D(C(i,j)) = M|C(i,j) - M(C(i,j))|^2 \quad (1)$$

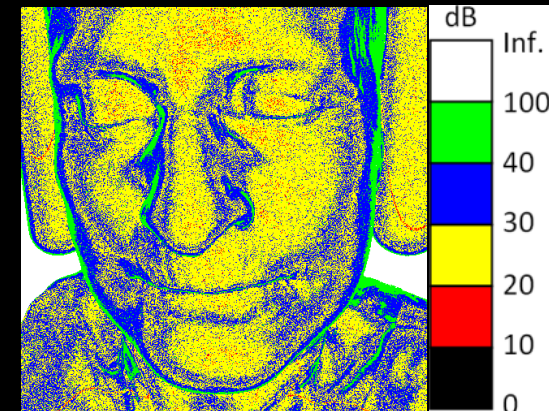
$$MSE = \frac{1}{N} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} D(I(i,j)) \quad (2)$$

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \quad (3)$$

$MSE$  – средняя квадратичная ошибка пикселя  
 $N$  – число пикселей, не принадлежащих фону;  
 $m, n$  – размеры окна вывода;



Результат рендеринга

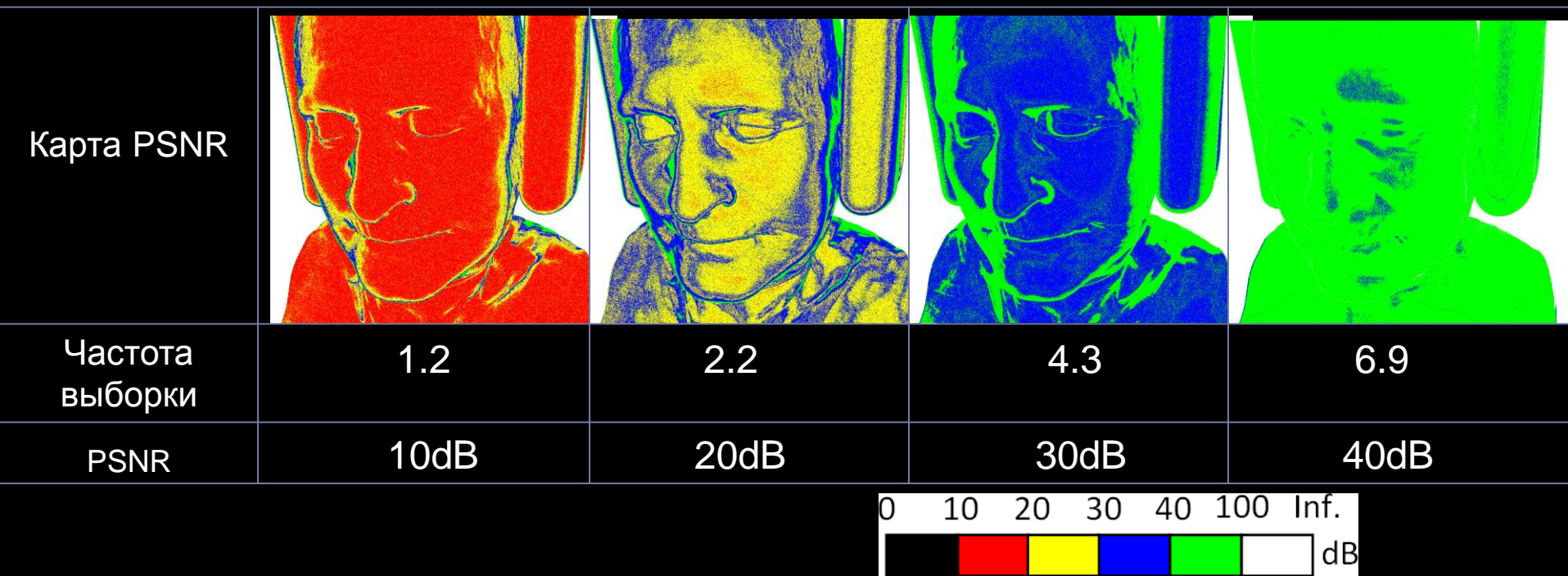


Карта PSNR

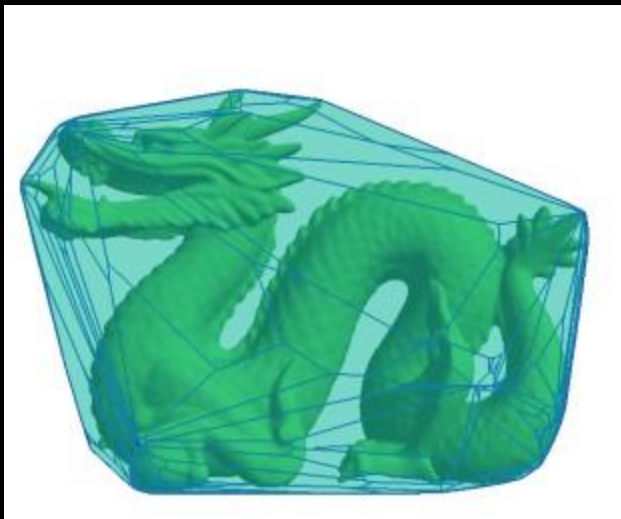
# Качество рендеринга при различных частотах выборки

Вне зависимости от метода рендеринга значение PSNR характеризует качество рендеринга:

- 1) PSNR  $\in$  [30dB,40dB] – приемлемое качество
- 2) PSNR > 40dB – улучшение качества практически незаметно
- 3) PSNR < 30dB – шум сильно заметен





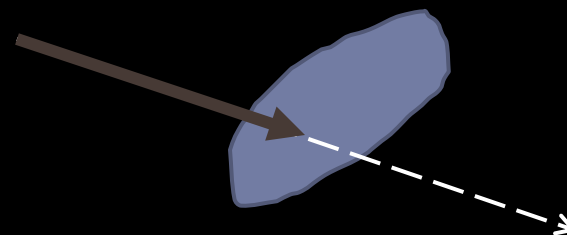


## Оптимизационные стратегии

# Возможные оптимизации

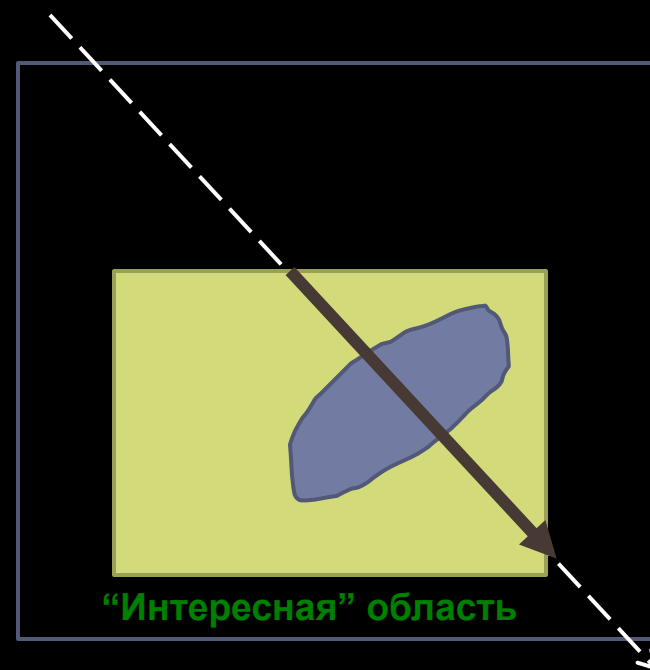
## ▶ Раннее завершение луча

- ▶ Сравнение накопленной непрозрачности с порогом



## ▶ Игнорирование пустых областей

- ▶ Хранить диапазоны значений блоков ячеек
- ▶ Хранить расстояние до ближайшего непустого вокселя
- ▶ Использовать список примитивов (параллелепипедов, сфер, полупространств), ограничивающих видимые данные

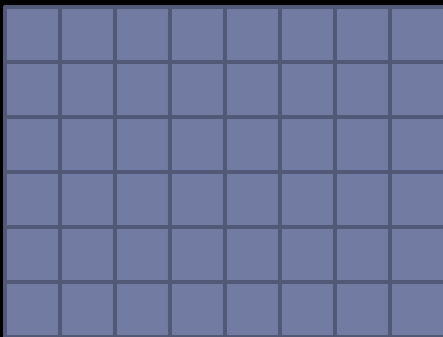


# Ускоряющая структура

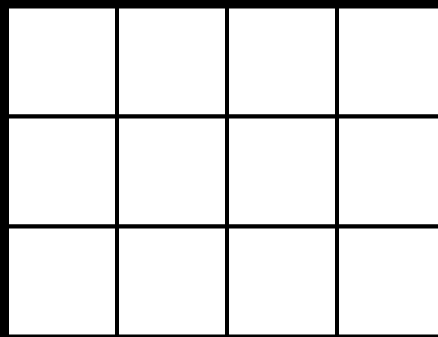
---

- ▶ Использование матрицы по размеру в несколько раз меньшей, чем исходная
- ▶ Элементы этой матрицы – оценка максимального и минимального значений данных, покрываемых макрочерчейкой

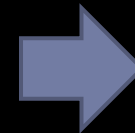
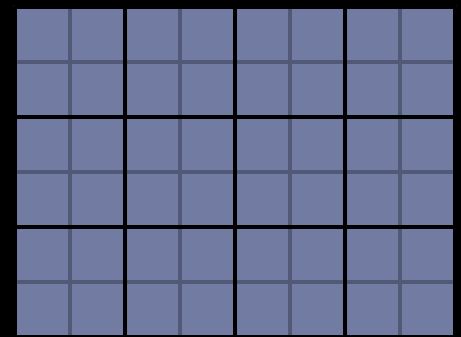
Исходные данные



Ускоряющая структура



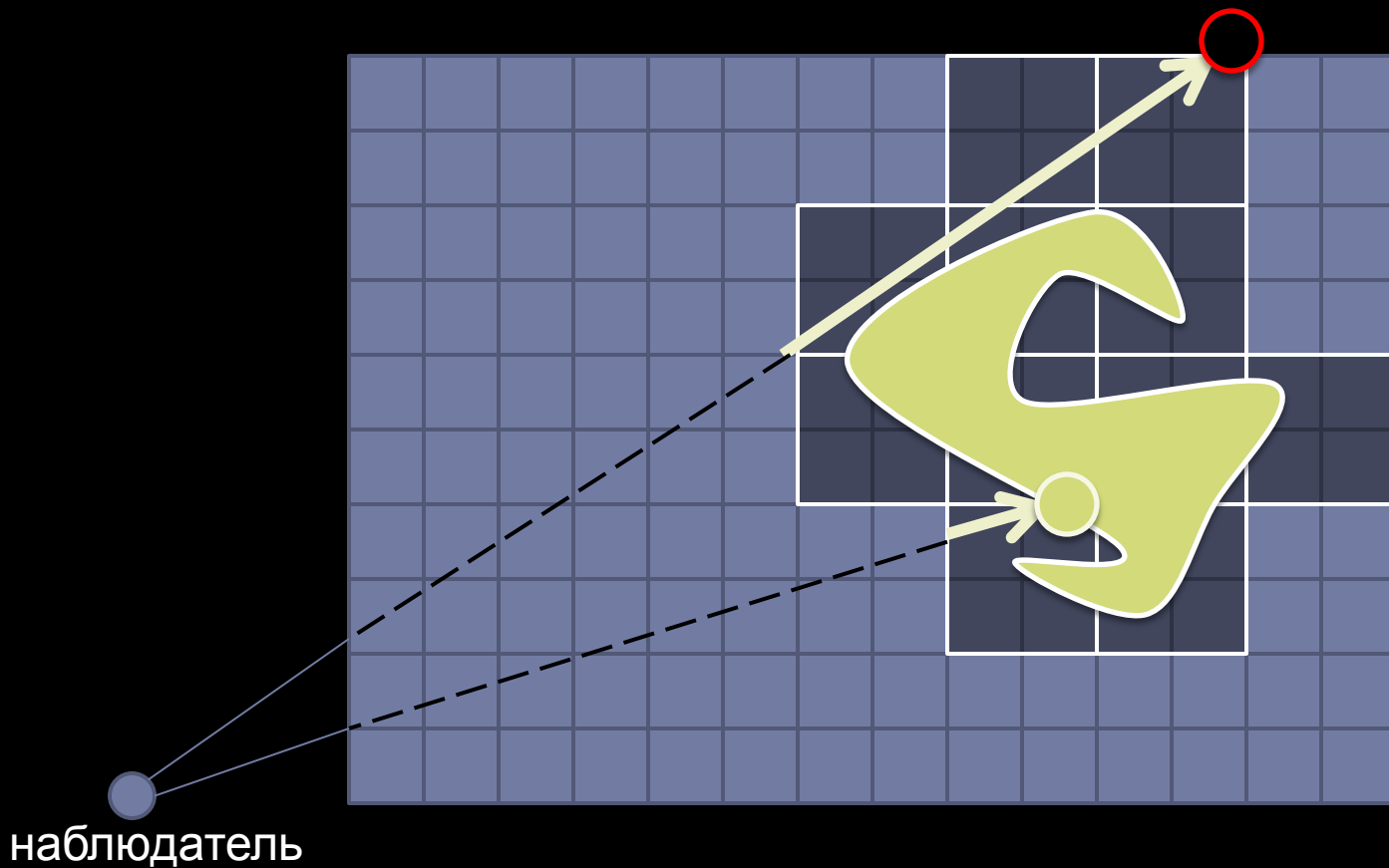
Иерархическая структура



# Ускоряющая структура

---

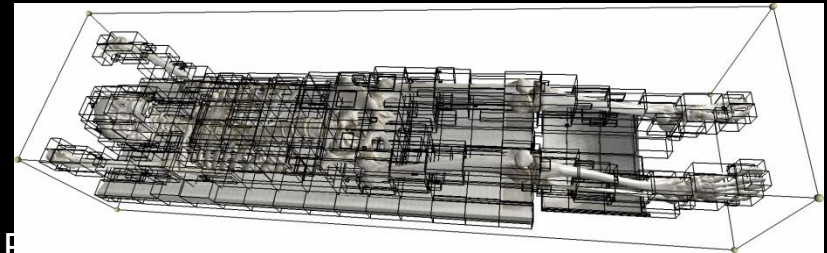
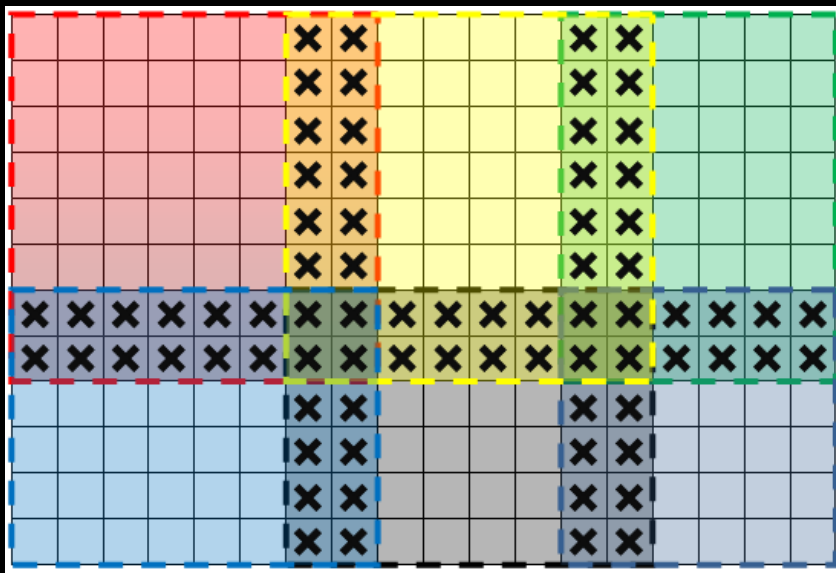
- ▶ Луч пропускает “неинтересные” области



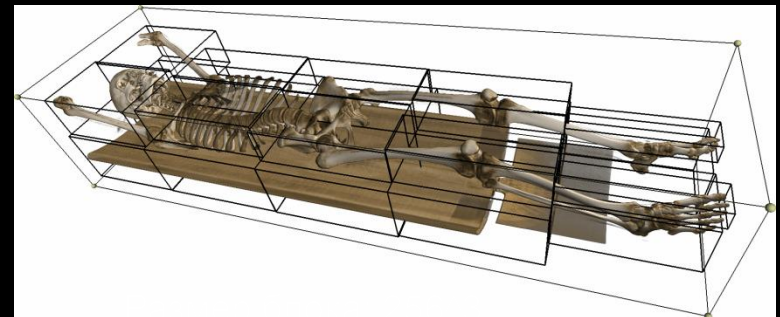
# Блочная декомпозиция пространственных данных

Модификация метода блочной декомпозиции данных в алгоритме Ray Casting, отличается:

- 1) оптимальной последовательностью обхода блоков;
- 2) использованием оптимизированных по объему вспомогательных структур метода пропуска пустых областей;
- 3) обеспечением возможности построения локального освещения и теней за счет перекрытия областей

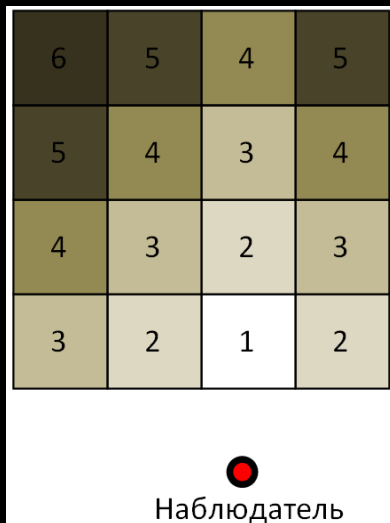


Размер блока: 64x64

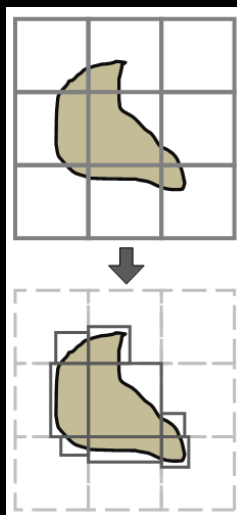




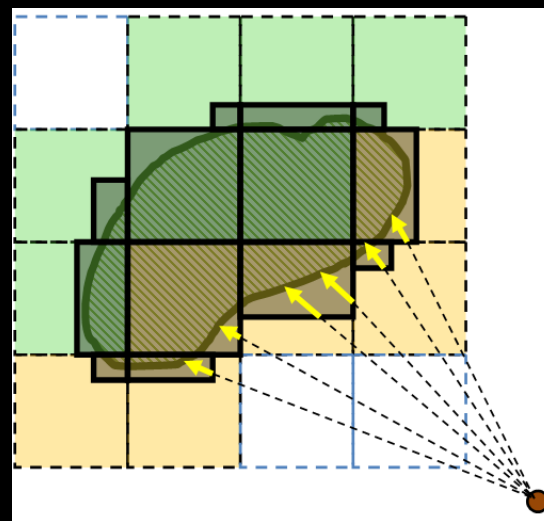
# Сокращение трассируемого объёма подгонкой ограничивающих блоков



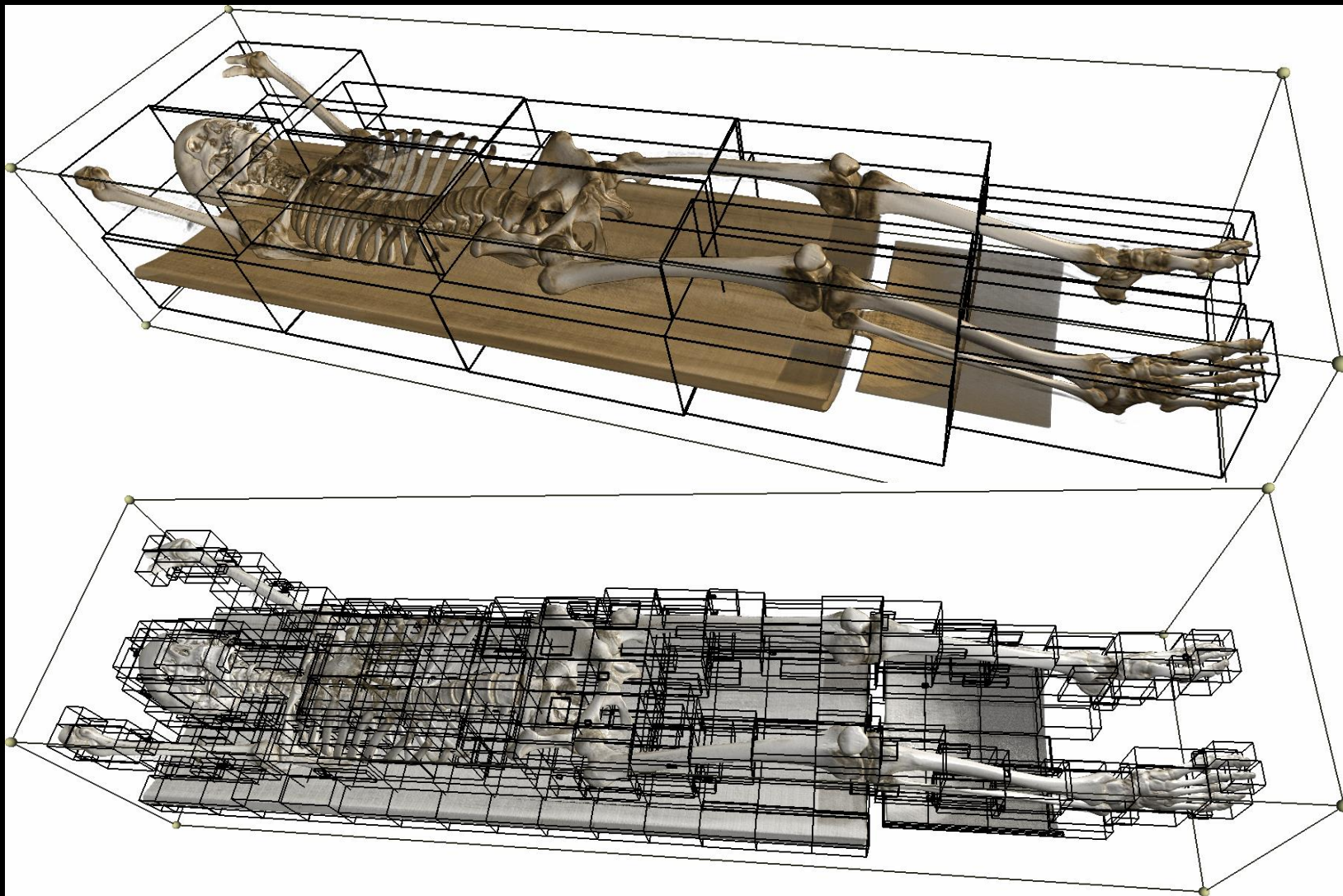
Оптимальная последовательность обхода блоков во время рендеринга



Для каждого блока данных заводится ограничивающий параллелепипед, который автоматически подгоняется под визуализируемые данные



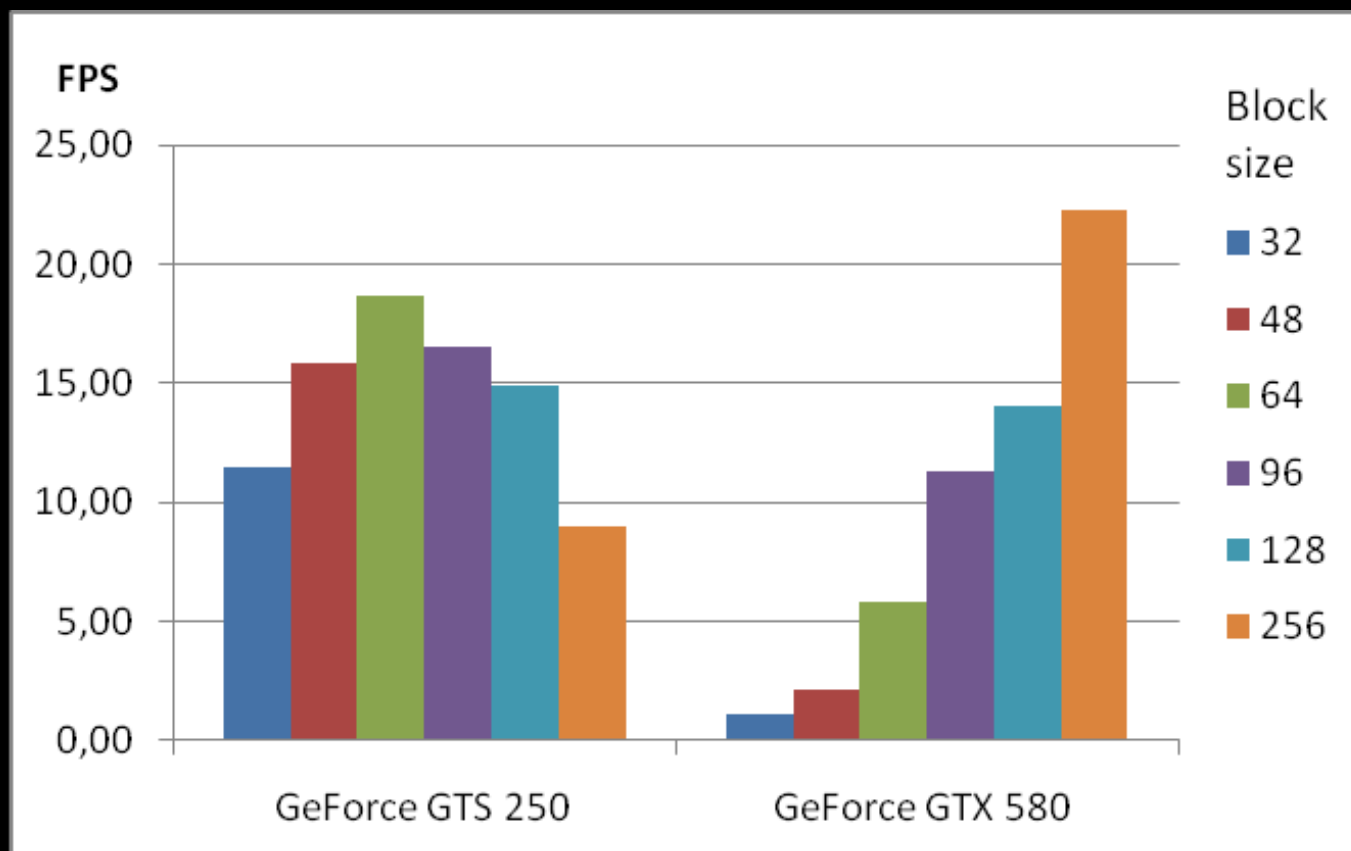
# Подгонка ограничивающих боксов



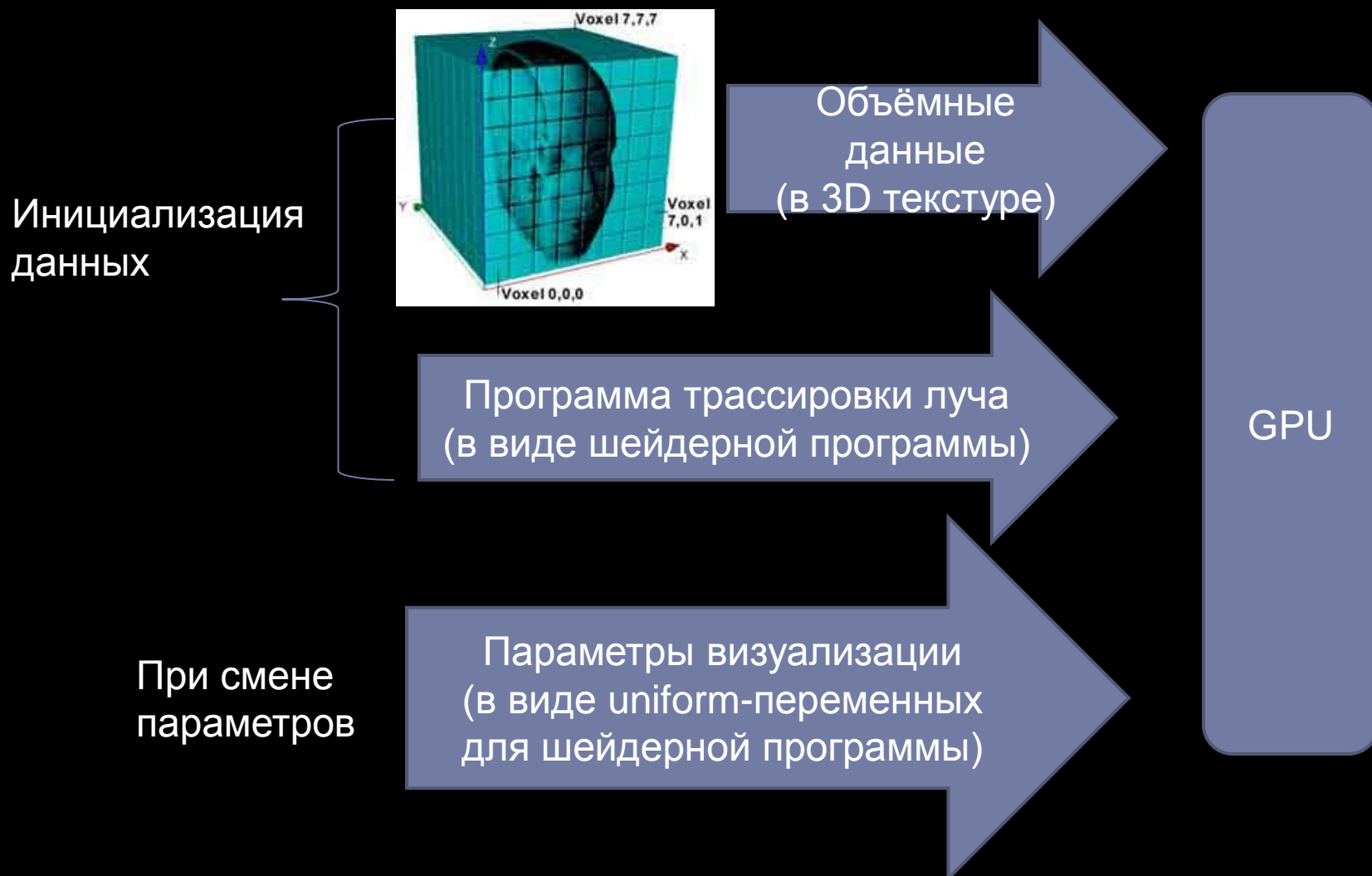
$256^3$

$64^3$

# Производительность рендеринга при различных размерах блоков: в зависимости от GPU оптимальны разный размер блока разбиения

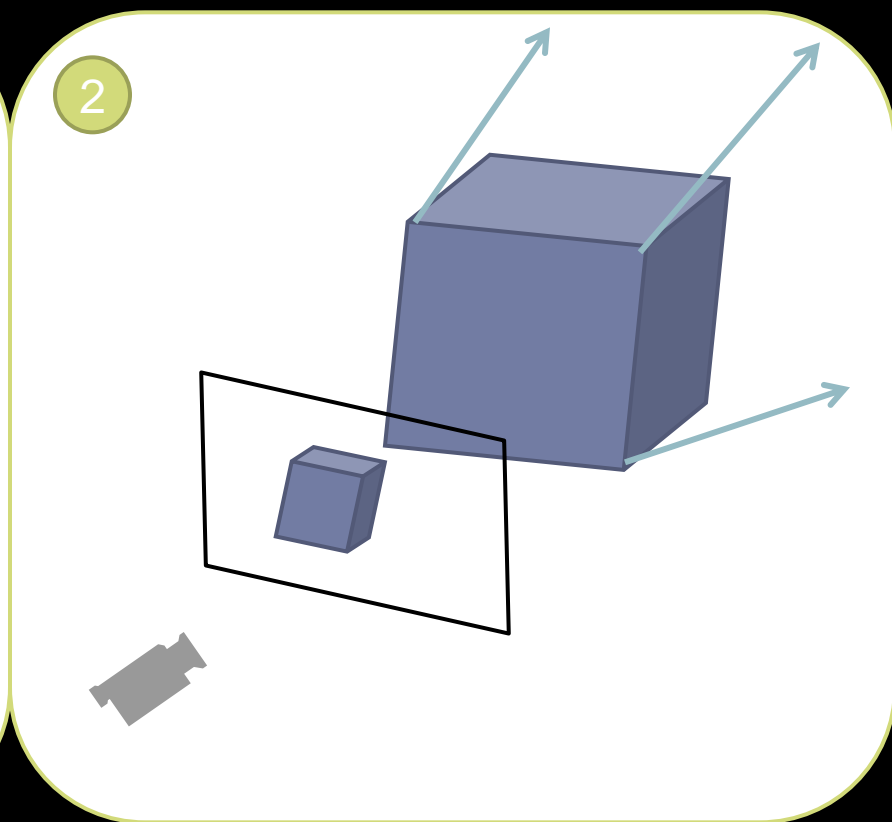
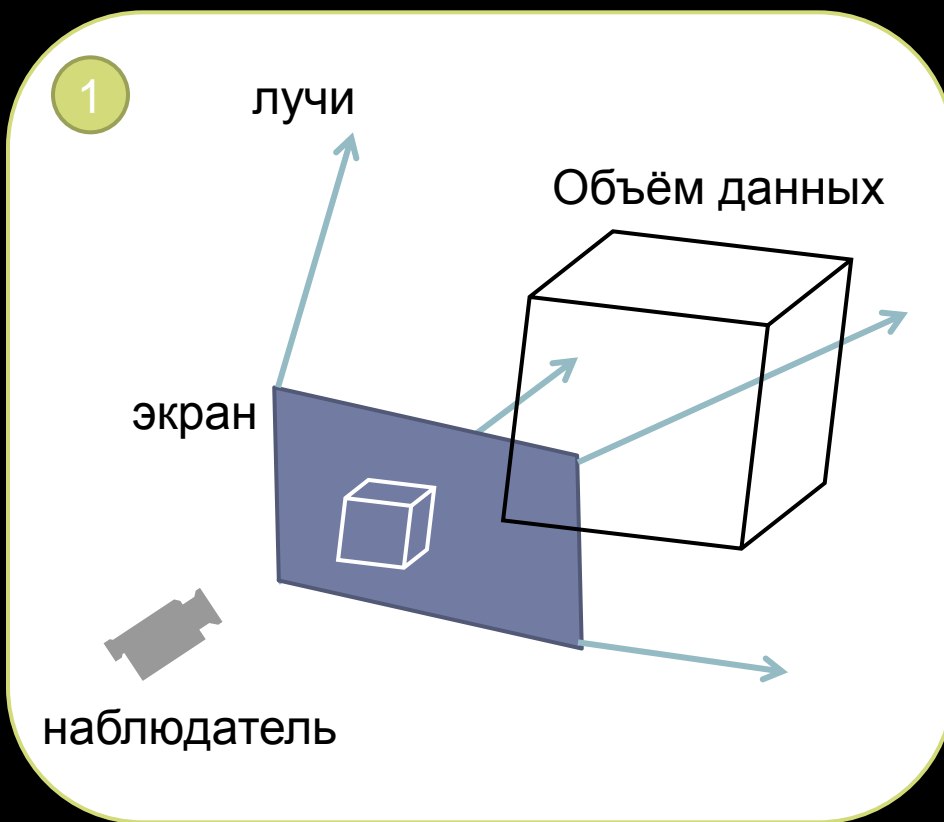


# Объёмный рендеринг на GPU



# Генерация лучей

- ▶ Генерация лучей. Интегрирование от картинной плоскости и от границы объема

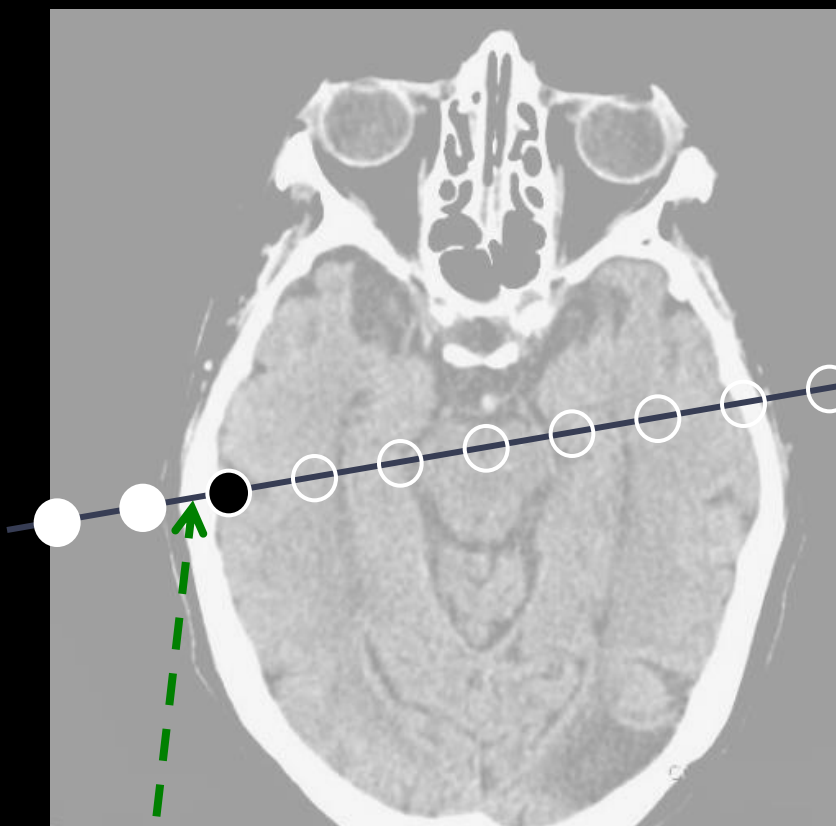






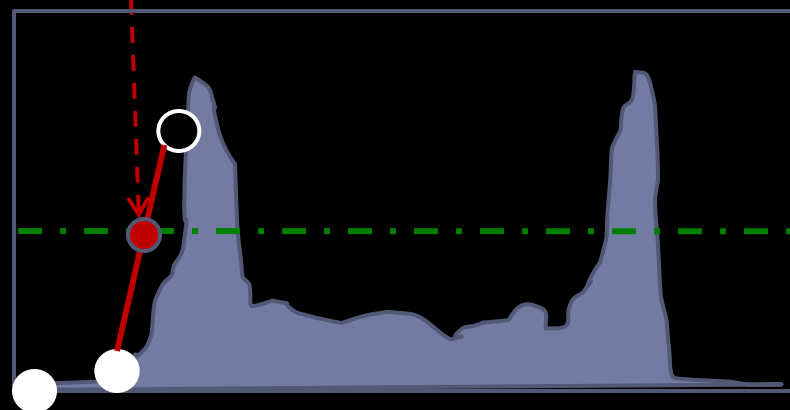
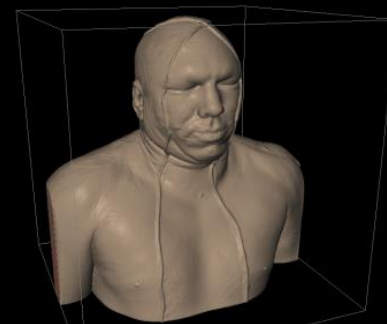
Различные алгоритмы накопления  
(интегрирования) цвета и  
непрозрачности вдоль луча

# Изоповерхность



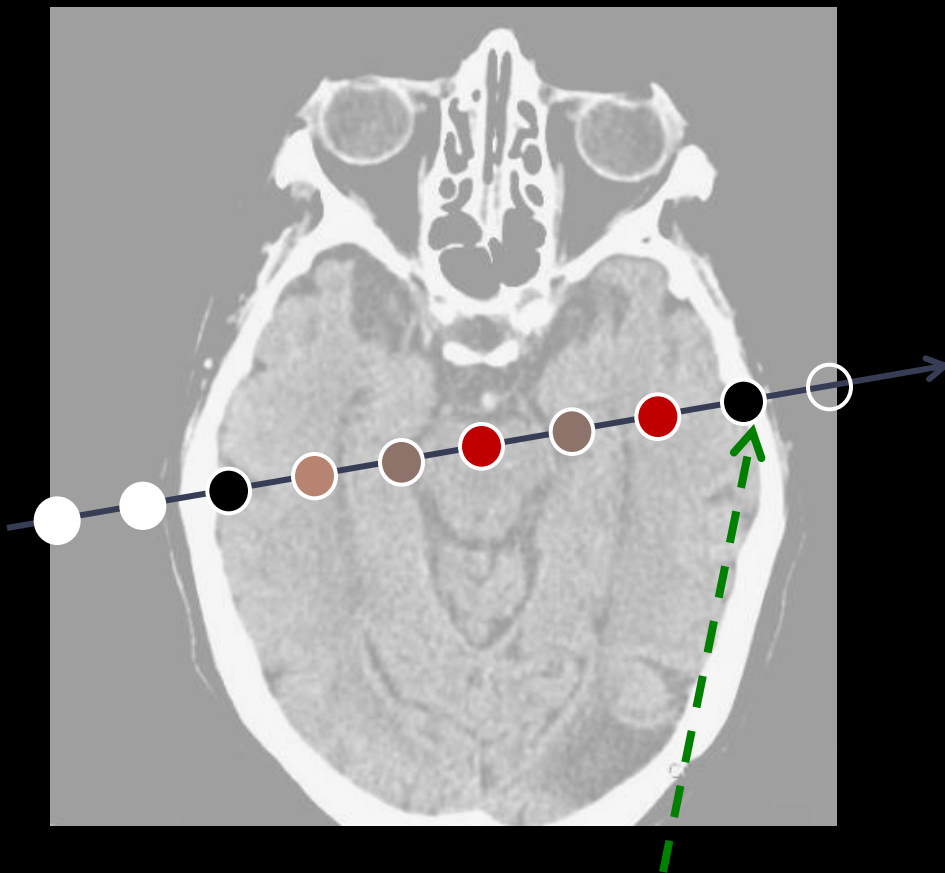
Отрезок, пересекающий изоповерхность

Приблизительное место соударения луча с изоповерхностью

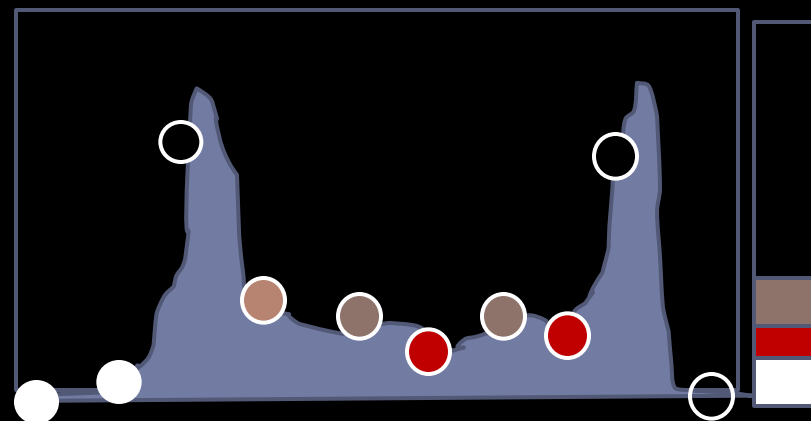
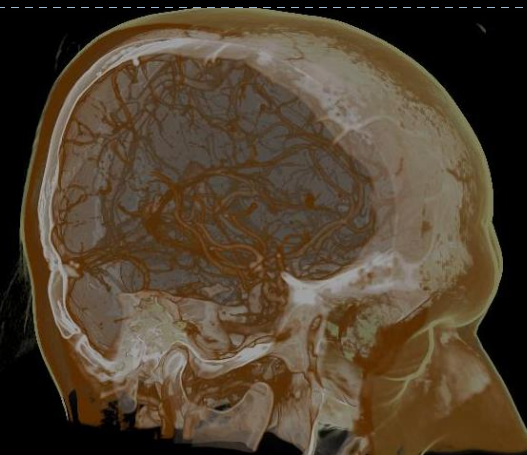


Значения вдоль луча

# Полупрозрачные среды

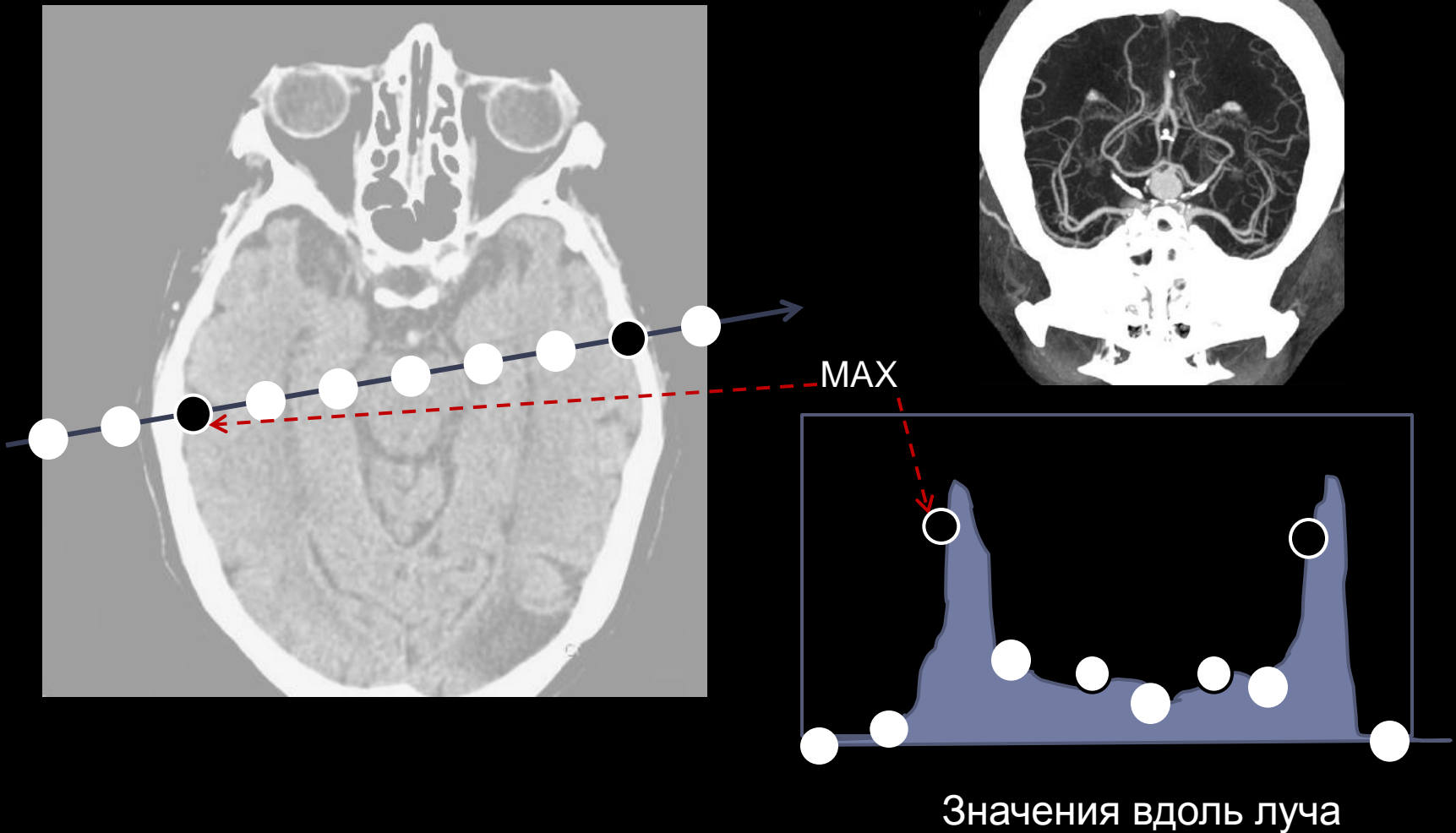


Накопление достаточной непрозрачности

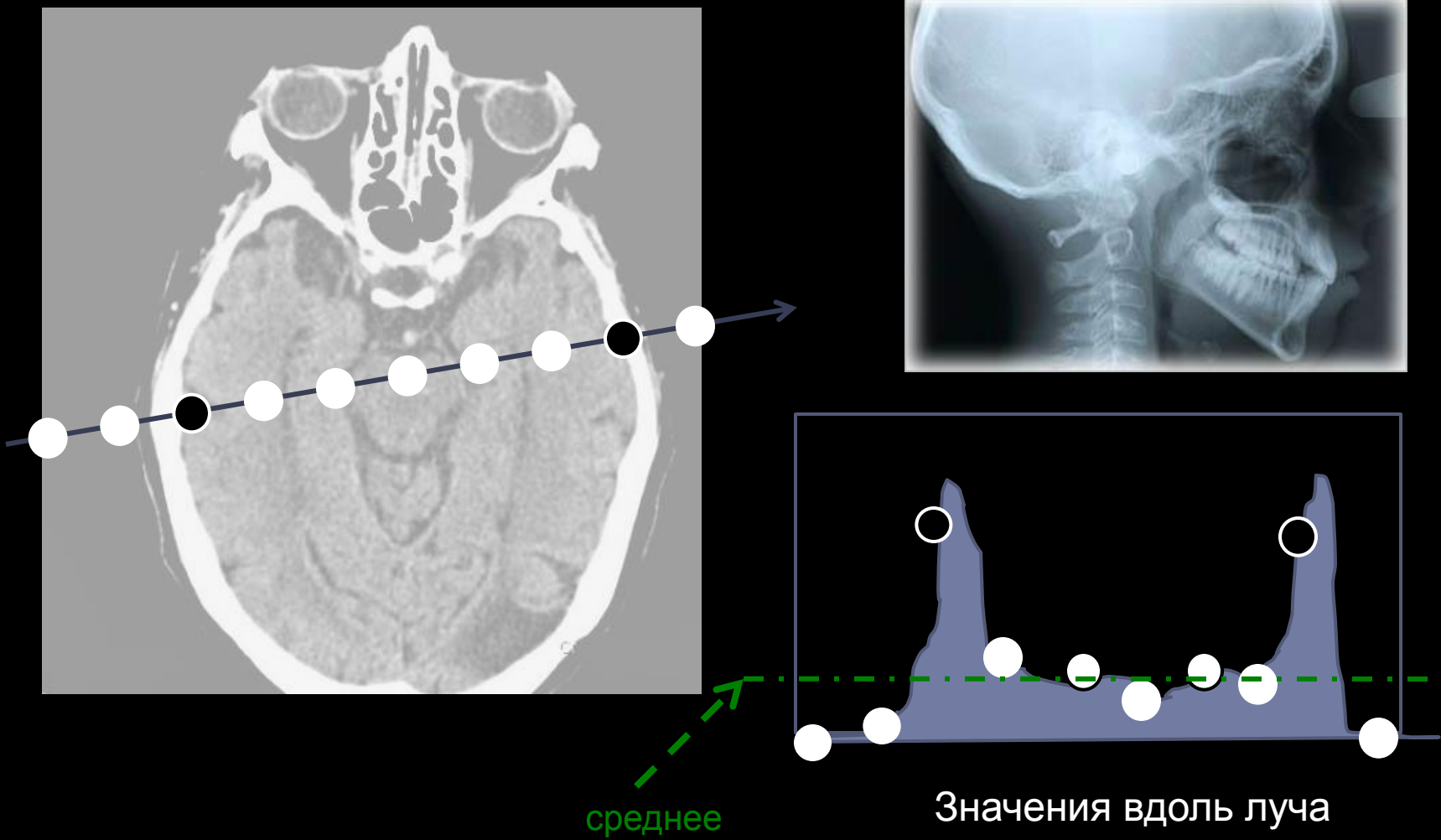


Значения вдоль луча

# MIP - проекция максимальной интенсивности



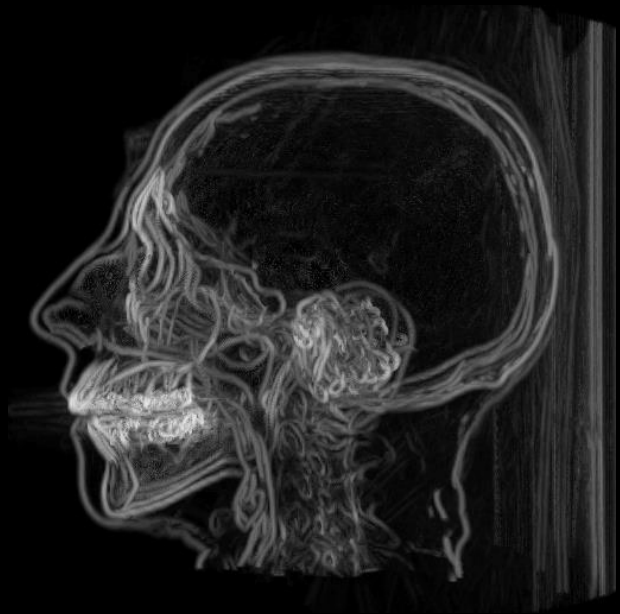
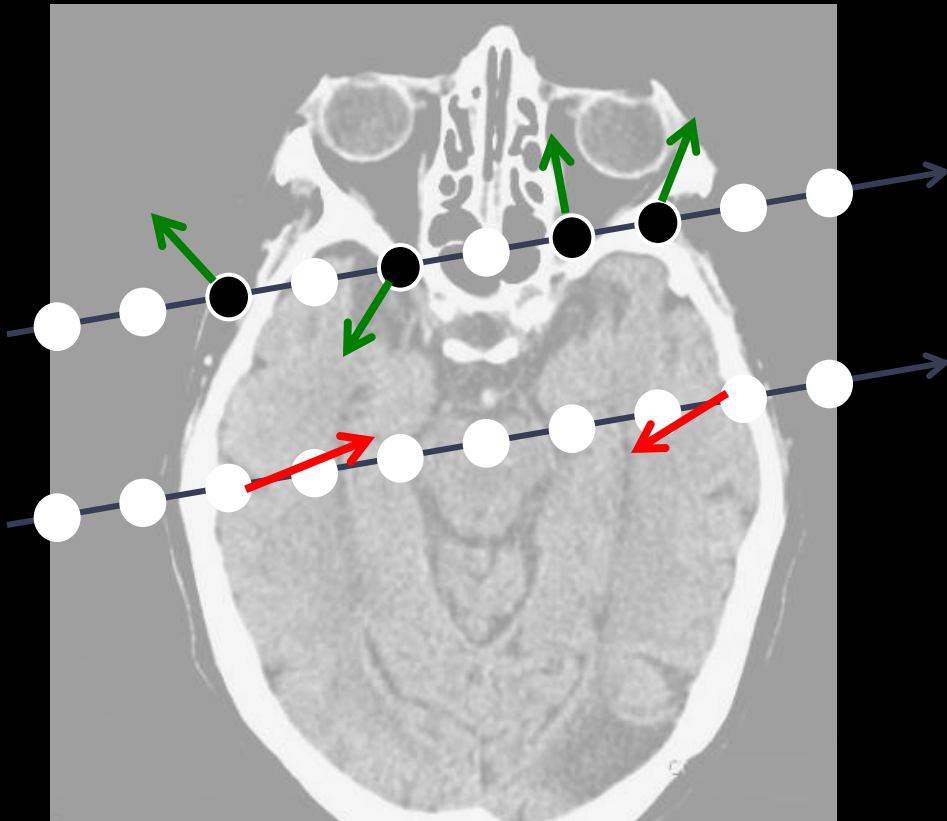
# X-Ray style (среднее значение, квазирентген)





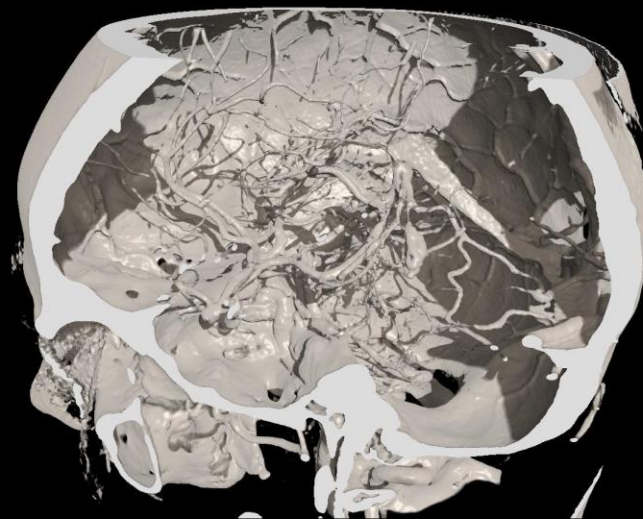
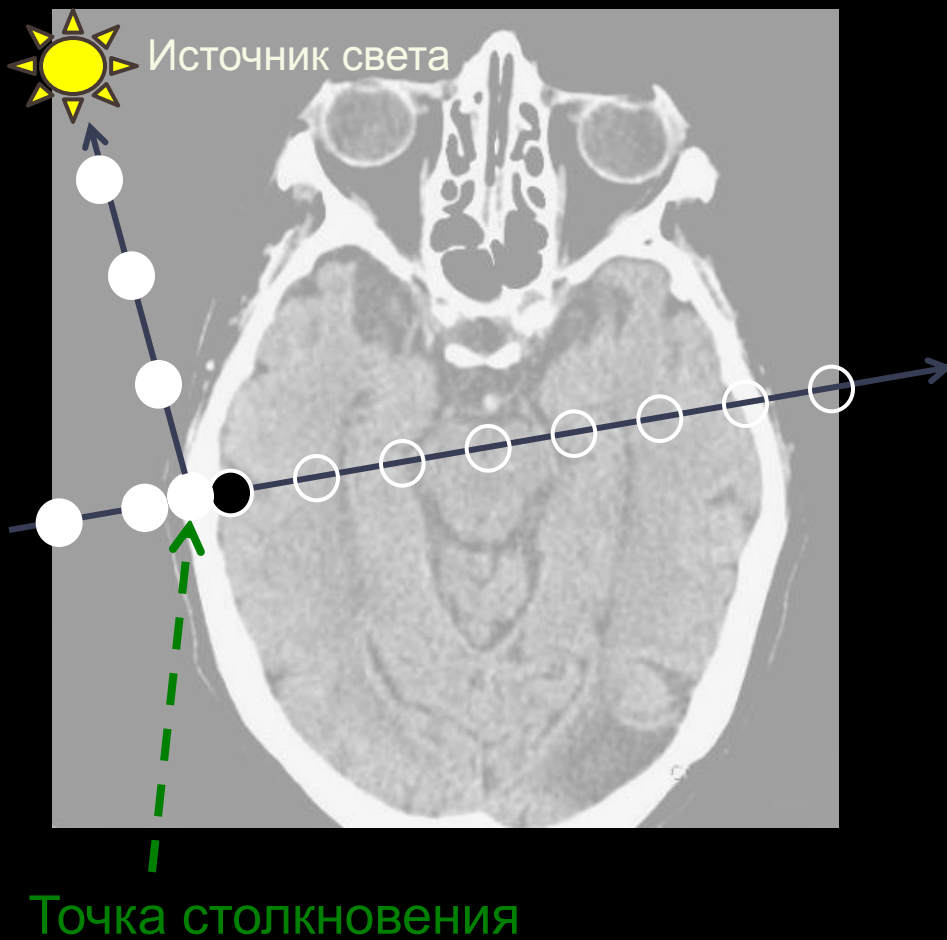
# Контуры

---



# Изоповерхность с тенью

---



---

**Лучевые методы в задаче глобальной  
освещенности в сценах виртуальной  
реальности.  
Массивная Монте-Карло-параллельность**

# Примеры. Интерактивная система виртуальной реальности на GPU.

Физически корректное глобальное освещение. 10.5 (до 30) млн. треугольников





# Примеры. Интерактивная система виртуальной реальности на GPU.

Физически корректное глобальное освещение. Рекордная производительность

Материалы: золото, медь, алюминий, стекло, пластик, жемчуг



## Производительность

Crytek Sponza.  
262K triangles, 1.8fps



Sibenik Cathedral.  
75K triangles, 2.1fps



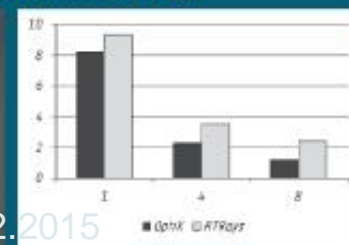
GPU NVIDIA GeForce 560, Ubuntu Linux 10.10 and CUDA Toolkit 4.0.  
Метод: стохастическая трассировка пути с глубиной 30.  
Разрешение: 1024 x 768. Усреднение: 500 кадров



**San-Miguel  
(Upper Floor),  
10500K,  
2.5-6 fps**

GPU NVIDIA  
GeForce 580  
Ubuntu Linux 10.10

Сравнение производительности с аналогами: NVIDIA OptiX  
NVIDIA GeForce GTX 680 4 Гб, Ubuntu Linux 12.



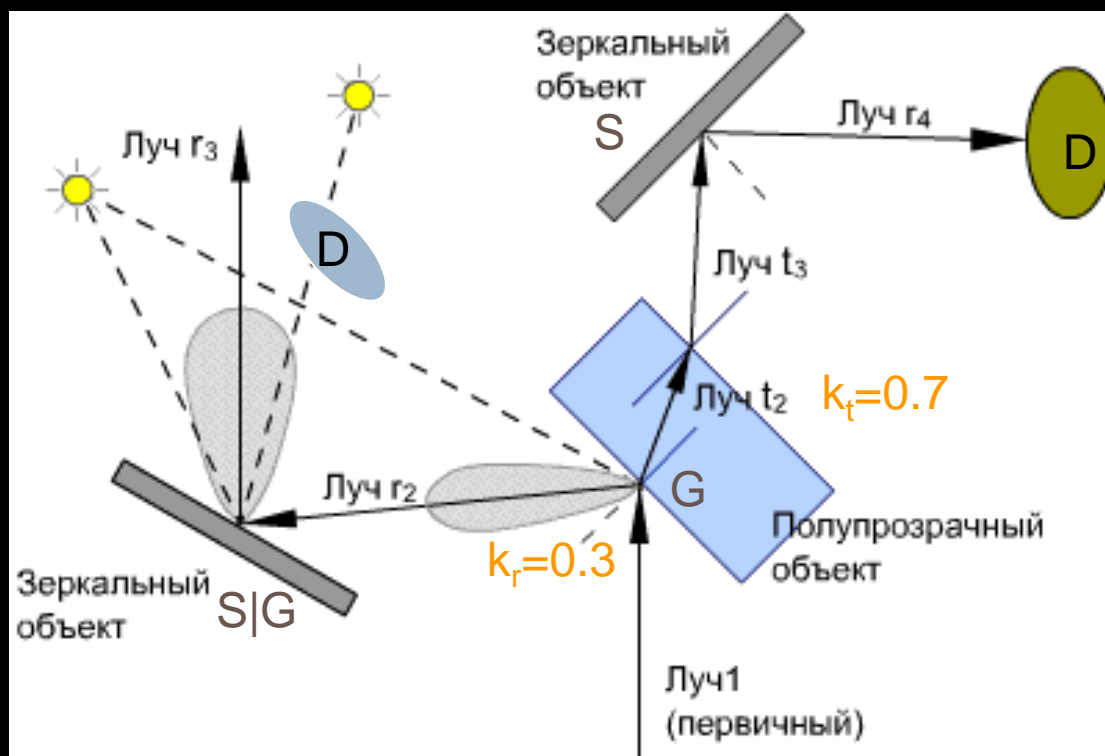
10.02.2015





# Рекурсивная процедура трассировки лучей

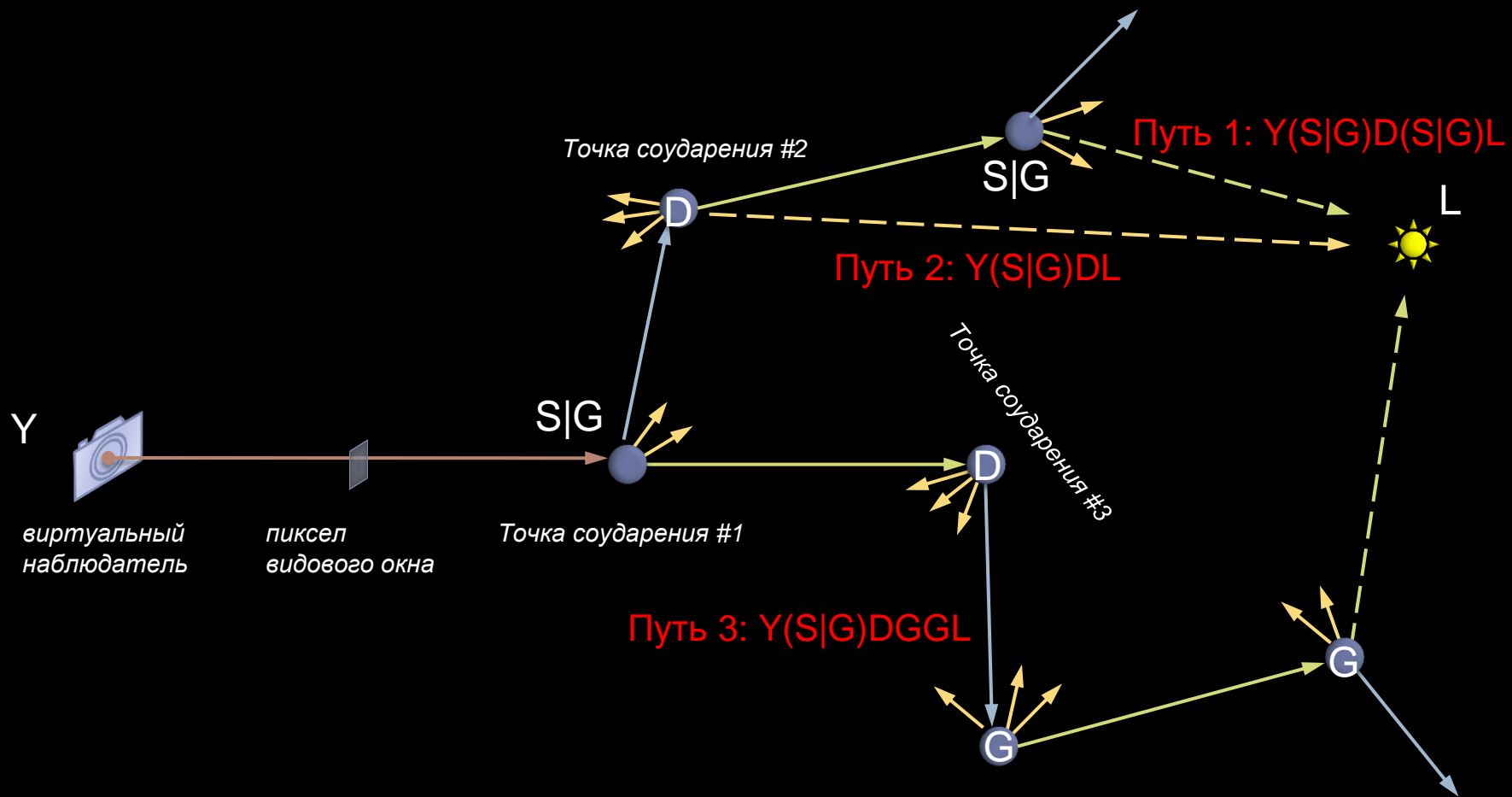
Обратная трассировка (трассировка Уиттеда). Рекурсивная процедура



Диффузно отражающий объект D

Пояснение :  
Луч t – преломленный;  
Луч r – отраженный;

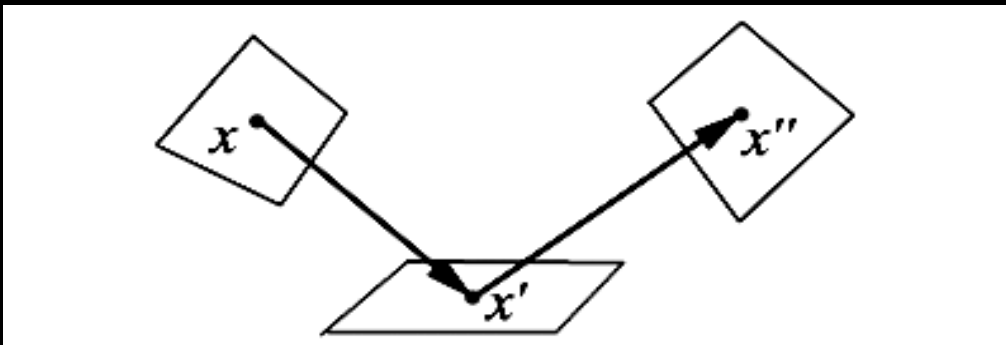
# Процесс трассировки путей



# Прямое и не прямое освещение. Основное уравнение освещенности

Световая энергия, которая излучается в точке  $x'$  в направлении другой точки  $x''$  - это сумма отраженной энергии и энергии излучаемой точкой самостоятельно. Должны быть учтены все лучи, которые пришли в эту точку от всех точек  $x \rightarrow$  интеграл.

$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x, x') G(x, x') dx$$



В этом уравнении функция  $L$  стоит как в правой, так и в левой части уравнения, причем в правой части под интегралом. Уравнения такого вида называются **уравнениями Фредгольма второго рода**, они не имеют аналитического решения (решаются численно).

Метод Монте-Карло интегрирования  $\rightarrow$  Шум: скорость сходимости –  $N^{-1/2}$ .  
В каждом испытании отслеживаем только один луч до конца  $\rightarrow$  однородность процесса  $\rightarrow$  массивная параллельность

# Примеры. Синтезированы интерактивной системой виртуальной реальности на графических процессорах



**Физически корректное  
глобальное освещение  
с рекордной  
производительностью**





---

**Открытые системы  
геометрического моделирования,  
инженерного и научного анализа и  
визуализации**



# Открытые системы геометрического моделирования, инженерного и научного анализа и визуализации

---

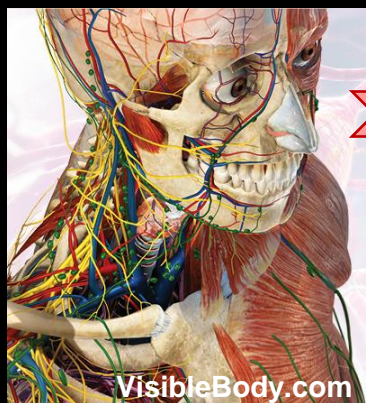
- Система трехмерного моделирования, сеточной декомпозиции и инженерного анализа [SALOME](#) ([localLink](#))
- Открытая система и технология для геометрического проектирования и инженерного анализа Open CASCADE Technology ([localLink](#))  
<http://www.opencascade.org/>  
[http://ru.wikipedia.org/wiki/Open\\_CASCADE\\_Technology](http://ru.wikipedia.org/wiki/Open_CASCADE_Technology)
- **OpenFOAM** ([англ. Open Source Field Operation And Manipulation CFD ToolBox](#)) — открытая интегрируемая платформа для [численного моделирования](#) задач [механики сплошных сред](#).
- Открытая система научно-технической визуализации **ParaView**  
<http://www.paraview.org/>
- **Visualization Toolkit (VTK)** is an open-source, freely available software system for 3D computer graphics, image processing and visualization  
[www.vtk.org](http://www.vtk.org)
- National Library of Medicine **Insight Segmentation and Registration Toolkit (ITK)**. <http://www.itk.org/>



# Новая цифровая 3D-медицина.

Диагностика. Образование

Состояние:  
разработка  
2014-2016



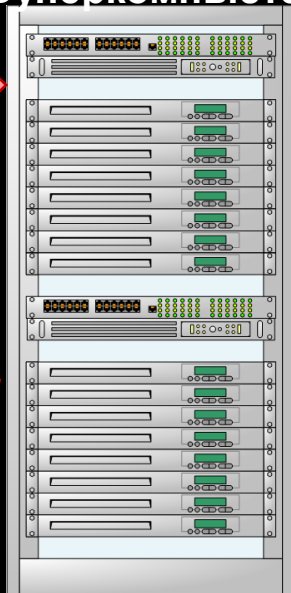
VisibleBody.com

Общая 3D модель



Персональная  
▶ томограмма

Суперкомпьютер



Индивидуальная  
3D модель

Виртуальный  
анатомический  
СТОЛ



Anatmage.com



Базы 3D Данных:  
Клиник региона  
Физических лиц  
Скрининга населения

Диагностические  
Базы Данных

Хранилища  
Больших Данных



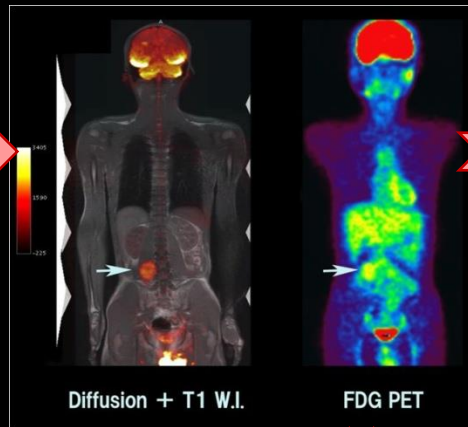
# Новая цифровая 3D-медицина.

Автоматическая диагностика.

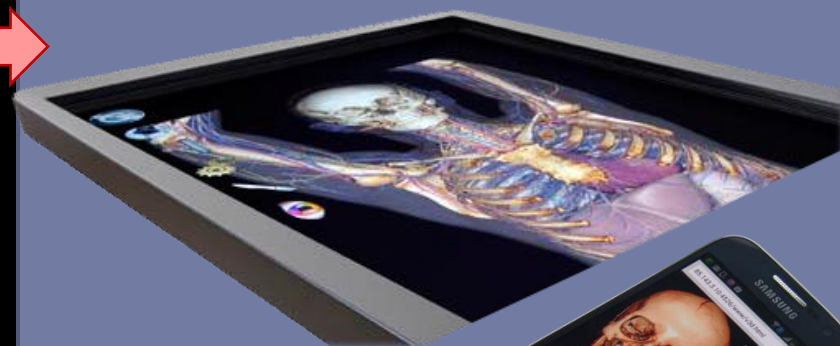
Состояние:  
разработка  
2014-2016

Суперкомпьютер

Предварительная  
автоматическая диагностика



Персональный кабинет  
врача -диагноста

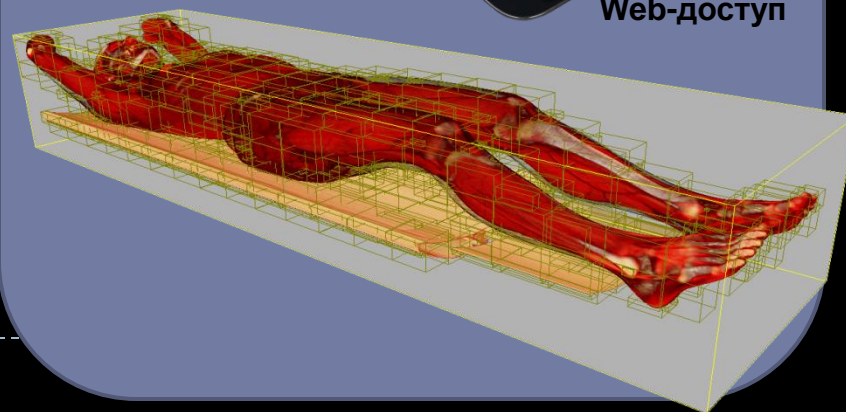


Индивидуальная  
3D модель,  
3D измерения



Web-доступ

Персональный кабинет  
пациента



Базы 3D Данных:  
Клиник региона  
Физических лиц  
Скрининга населения

Диагностические  
Базы Данных

Хранилища  
Больших Данных



# Новые 3D технологии в области медицины и здравоохранения.

## 3D модели анатомии органов и систем человека.

### 3D Avatar projects. Zygote Group Inc. – Human Anatomy Collection

[http://www.3dscience.com/3D\\_Models/Human\\_Anatomy/Collections/Male-Female\\_Collection.php](http://www.3dscience.com/3D_Models/Human_Anatomy/Collections/Male-Female_Collection.php)

#### 3D MODELS

#### HUMAN ANATOMY

#### COLLECTIONS

Male systems

Female systems

Brain

Heart

Infant

Reproductive

Sensory

Skeleton

Skin

Teeth

Urinary

#### BIOLOGY

#### MEDICAL

#### SOLID MODELS

#### HUMAN FACTORS

#### CASE STUDIES

#### UPGRADES

#### RESOURCES

#### NEWSLETTER

#### CLIP ART

#### FAQ

#### CUSTOM SERVICES

#### COMPANY



#### Male & Female Anatomy Collection 5.0

Polygons: Yes

UV Coordinates: available

Textures: available

Grouping: Yes

Download Size: (geometry) 100MB - 200MB

Download Size: (with maps) 340MB - 800MB

Price:

**\$20,995 without Textures**

**\$24,995 with Textures**

Please call us at 801.765.4141 ext. 102 or use our [Contact Us](#) form to place an order.

Formats:

» For discounted package rates please contact us.

[VIEW 5.0 UPGRADE DETAILS](#)

» [VIEW DETAILED PRODUCT DESCRIPTION](#)

» [VIEW INCLUDED FEMALE SYSTEMS](#)

» [VIEW INCLUDED MALE SYSTEMS](#)